



LTPDA Unit Test Report S2-AEI-TN-XXXX

prepared by	M Hewitson
checked by	
Issue	1
Revision	1
Status	Draft
Number of pages	1094
date of issue	April 28, 2011
approved by	

This document and all parts of it are confidential. Any distribution is prohibited without written authorisation from AEI and UNITN.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



Distribution List

Name	Company/ Institute
	AEI Hannover



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

Contents

1	Introduction	11
	1.1 Version	11
2	Results	11

List of Tables

1	Unit tests for ao/abs
2	Unit tests for ao/acos
3	Unit tests for ao/and
4	Unit tests for ao/angle
5	Unit tests for ao/ao
6	Unit tests for ao/asin
7	Unit tests for ao/atan
8	Unit tests for ao/atan2
9	Unit tests for ao/average
10	Unit tests for bin/data
11	Unit tests for ao/cat
12	Unit tests for ao/char. 92
13	Unit tests for ao/cohere
14	Unit tests for ao/complex
15	Unit tests for ao/compute
16	Unit tests for ao/conj
17	Unit tests for ao/conv
18	Unit tests for ao/convert
19	Unit tests for ao/copy
20	Unit tests for ao/cos
21	Unit tests for ao/cov
22	Unit tests for ao/cpsd
23	Unit tests for ao/created
24	Unit tests for ao/creator
25	Unit tests for ao/ctranspose
26	Unit tests for ao/delay
27	Unit tests for ao/demux
28	Unit tests for ao/det
29	Unit tests for ao/detrend
30	Unit tests for ao/dft
31	Unit tests for ao/diag
32	Unit tests for ao/diff
33	Unit tests for ao/display
34	Unit tests for ao/dopplercorr
35	Unit tests for ao/downsample
36	Unit tests for ao/dropduplicates
37	Unit tests for ao/dsmean
38	Unit tests for ao/eig
39	Unit tests for ao/eq
40	Unit tests for ao/exp
41	Unit tests for ao/export
42	Unit tests for ao/fft



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



43	Unit tests for ao/fftfilt.	153
44	Unit tests for ao/filtSubtract.	155
45	Unit tests for ao/filter.	159
46	Unit tests for ao/filtfilt.	161
47	Unit tests for ao/find	163
48	Unit tests for ao/firwhiten.	165
49	Unit tests for ao/fixfs.	168
50	Unit tests for ao/fs.	169
51	Unit tests for ao/ge	170
52	Unit tests for ao/get.	171
53	Unit tests for ao/gt.	172
54	Unit tests for ao/heterodyne	174
55	Unit tests for ao/hist.	176
56	Unit tests for ao/ifft.	178
57	Unit tests for ao/imag.	180
58	Unit tests for ao/index.	182
59	Unit tests for ao/integrate.	184
60	Unit tests for ao/interp.	186
61	Unit tests for ao/interprissing	189
62	Unit tests for ao/inv	191
63	Unit tests for ao/isprop	193
64	Unit tests for join/fsdata	195
65	Unit tests for join/tsdata	197
66	Unit tests for ao/loohere	201
67	Unit tests for ad/lepsd	201
68	Unit tests for ao/le	200
60	Unit tests for a /len	204
70	Unit tests for ad/linSubtract	$200 \\ 207$
70	Unit tests for ad/information and a second s	201
71	Unit tests for ad/Infom.	209
14 73	Unit tests for ac/leadebi	211
73	Unit tests for ad/log	212 914
75	Unit tests for ac/log10	214
10 76	Unit tests for ac/log10	210
70	Unit tests for ac/laser	221
79	Unit tests for ac/lt	223
10 70	Unit tests for ac/ltfc	224
19	Unit tests for ac/mar	220
8U 01	Unit tests for a / max.	228
81	Unit tests for a / mcmc.	230
82		231
83	Unit tests for ao/mean.	233
84	Unit tests for ao/median.	235
85	Unit tests for ao/min.	237
86	Unit tests for ao/minus.	295
87	Unit tests for ao/mode	297
88	Unit tests for ao/mpower.	299
89	Unit tests for ao/mrdivide	357
90	Unit tests for ao/mtimes	358
91	Unit tests for ao/ne.	359
92	Unit tests for ao/noisegen1D.	361
93	Unit tests for ao/noisegen2D.	363
94	Unit tests for ao/norm.	365



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



95	Unit tests for ao/offset	367
96	Unit tests for ao/or	425
97	Unit tests for ao/phase	427
98	Unit tests for ao/plus.	485
99	Unit tests for ao/polyfit.	487
100	Unit tests for ao/power.	489
101	Unit tests for ao/psd	504
102	Unit tests for ao/rdivide	562
103	Unit tests for ao/real	564
104	Unit tests for ao/rebuild.	565
105	Unit tests for ao/resample	567
106	Unit tests for ao/rms	569
107	Unit tests for ao/round.	571
108	Unit tests for ao/sDomainFit.	573
109	Unit tests for ao/save.	576
110	Unit tests for ao/scale.	578
111	Unit tests for ao/search.	580
112	Unit tests for ao/select	583
113	Unit tests for ao/setDescription.	585
114	Unit tests for ao/setFs	587
115	Unit tests for ao/setName.	589
116	Unit tests for ao/setPlotinfo.	591
117	Unit tests for ao/setT0.	593
118	Unit tests for ao/setX.	595
119	Unit tests for ao/setXY.	597
120	Unit tests for ao/setXunits.	599
121	Unit tests for ao/setY.	601
122	Unit tests for ao/setYunits	603
123	Unit tests for ao/sign	605
124	Unit tests for ao/simplifyYunits	607
125	Unit tests for ao/sin.	609
126	Unit tests for ao/smoother	611
127	Unit tests for ao/sort	613
128	Unit tests for ao/spectrogram.	615
129	Unit tests for split/chunks.	617
130	Unit tests for split/interval.	620
131	Unit tests for split/samples.	622
132	Unit tests for times/frequ.	625
133	Unit tests for ao/sqrt	627
134	Unit tests for ao/std	629
135	Unit tests for ao/string	630
136	Unit tests for ao/sum.	632
137	Unit tests for ao/sumjoin.	634
138	Unit tests for ao/svd	636
139	Unit tests for $ao/t0$	637
140	Unit tests for ao/tan	639
141	Unit tests for ao/tfe.	542
142	Unit tests for ao/times	700
143	Unit tests for ao/timeshift	702
144	Unit tests for ao/transpose.	704
145	Unit tests for ao/type	705
146	Unit tests for ao/uminus	707



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

147	Unit tests for ao/unwrap	709
148	Unit tests for ao/upsample.	711
149	Unit tests for ao/var	713
150	Unit tests for ao/whiten1D.	715
151	Unit tests for ao/whiten2D.	717
152	Unit tests for ao/x	718
153	Unit tests for ao/xcorr	721
154	Unit tests for ao/xunits.	723
155	Unit tests for ao/y	724
156	Unit tests for ao/yunits.	725
157	Unit tests for ao/zDomainFit.	727
158	Unit tests for ao/zeropad.	729
159	Unit tests for collection/collection.	732
160	Unit tests for collection/copy.	733
161	Unit tests for collection/loadobj	734
162	Unit tests for filterbank/copy.	735
163	Unit tests for filterbank/filterbank. 7	737
164	Unit tests for filterbank/loadobj.	738
165	Unit tests for matrix/char	739
166	Unit tests for matrix/copy	740
167	Unit tests for matrix/ctranspose.	742
168	Unit tests for matrix/display.	743
169	Unit tests for matrix/loadobj.	744
170	Unit tests for matrix/matrix	747
171	Unit tests for matrix/ncols	748
172	Unit tests for matrix/nrows.	749
173	Unit tests for matrix/osize	750
174	Unit tests for matrix/setObjs.	752
175	Unit tests for matrix/transpose	754
176	Unit tests for mfir/char.	755
177	Unit tests for mfir/copy.	756
178	Unit tests for mfir/created	758
179	Unit tests for mfir/creator.	760
180	Unit tests for mfir/display.	761
181	Unit tests for mfir/eq	763
182	Unit tests for mfir/get.	764
183	Jnit tests for mfir/index	766
184	Jnit tests for mfir/isprop.	768
185	Jnit tests for mfir/loadobj	769
186	Jnit tests for mfir/mfir.	774
187	Jnit tests for mfir/ne.	775
188	Jnit tests for mfir/rebuild	776
189	Unit tests for mfir/redesign.	778
190	Jnit tests for mfir/resp	781
191	Unit tests for mir/save.	<i>(</i> 83
192	Unit tests for mfir/setHistout.	<i>(</i> 85
193	Unit tests for mfir/setlunits.	787
194	Unit tests for mfir/setName	789
195	Unit tests for mfir/setOunits.	791
196	Unit tests for mfir/string	792
197	Unit tests for mfir/type	793
198	Unit tests for miir/char.	794



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



199 Unit tests for miir/copy	95
200 Unit tests for miir/created	97
201 Unit tests for miir/creator	99
202 Unit tests for miir/display	00
203 Unit tests for miir/eq	02
204 Unit tests for miir/get	03
205 Unit tests for miir/index	05
206 Unit tests for miir/isprop	07
207 Unit tests for miir/loadobj	08
208 Unit tests for miir/miir	13
209 Unit tests for miir/ne	14
210 Unit tests for miir/rebuild	15
211 Unit tests for miir/redesign	17
212 Unit tests for miir/resp	20
213 Unit tests for miir/save	22
214 Unit tests for miir/setHistin	23
215 Unit tests for miir/setHistout.	25
216 Unit tests for miir/setIunits.	27
217 Unit tests for miir/setName	29
218 Unit tests for miir/setOunits.	31
219 Unit tests for mir/string	32
220 Unit tests for miir/type	33
221 Unit tests for parfrac/char	34
222 Unit tests for parfrac/copy	35
223 Unit tests for parfrac/created	37
224 Unit tests for parfrac/creator	39
225 Unit tests for parfrac/display	40
226 Unit tests for parfrac/eq	42
227 Unit tests for parfrac/get.	43
228 Unit tests for parfrac/getlowerFreq	44
229 Unit tests for parfrac/getupperFreq	45
230 Unit tests for parfrac/index.	47
231 Unit tests for parfrac/isprop.	49
232 Unit tests for parfrac/loadobi.	50
233 Unit tests for parfrac/ne	51
234 Unit tests for parfrac/parfrac.	55
235 Unit tests for parfrac/rebuild.	56
236 Unit tests for parfrac/resp	59
237 Unit tests for parfrac/save.	61
238 Unit tests for parfrac/setJunits.	63
239 Unit tests for parfrac/setName.	65
240 Unit tests for parfrac/setOunits.	67
241 Unit tests for parfrac/string	68
242 Unit tests for parfrac/type	69
243 Unit tests for pest/copy 8	70
244 Unit tests for pest/loadobi	71
245 Unit tests for pest/pest.	73
246 Unit tests for pest/setChain 8	75
247 Unit tests for pest/setChi2	77
248 Unit tests for pest/setCorr 8	70
240 Unit tests for pest/setCov	81
250 Unit tests for post/setDof	85
	00



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



251 Unit tests for pest/setDy	85
252 Unit tests for pest/setModels	87
253 Unit tests for pest/setNames	89
254 Unit tests for pest/setPdf	91
255 Unit tests for pest/setY	93
256 Unit tests for pest/setYunits	95
257 Unit tests for plist/append	97
258 Unit tests for plist/char	98
259 Unit tests for plist/combine	00
260 Unit tests for plist/copy	01
261 Unit tests for plist/display	02
262 Unit tests for plist/eq	03
263 Unit tests for plist/find	04
264 Unit tests for plist/get	05
265 Unit tests for plist/isparam	06
266 Unit tests for plist/isprop	07
267 Unit tests for plist/loadobj	08
268 Unit tests for plist/ne	09
269 Unit tests for plist/nparams	10
270 Unit tests for plist/parse	11
271 Unit tests for plist/plist	13
272 Unit tests for plist/plist2cmds	14
273 Unit tests for plist/pset	16
274 Unit tests for plist/remove	18
275 Unit tests for plist/save	20
276 Unit tests for plist/setName	22
277 Unit tests for plist/string	23
278 Unit tests for pzmodel/char	24
279 Unit tests for pzmodel/copy	25
280 Unit tests for pzmodel/created	27
281 Unit tests for pzmodel/creator	29
282 Unit tests for pzmodel/display	30
283 Unit tests for pzmodel/eq	32
284 Unit tests for pzmodel/get	33
285 Unit tests for pzmodel/getlowerFreq	34
286 Unit tests for pzmodel/getupperFreq	35
287 Unit tests for pzmodel/index	37
288 Unit tests for pzmodel/isprop	39
289 Unit tests for pzmodel/loadobj	40
290 Unit tests for pzmodel/mrdivide	42
291 Unit tests for pzmodel/mtimes	44
292 Unit tests for pzmodel/ne	45
293 Unit tests for pzmodel/pzmodel	49
294 Unit tests for principal (reliable)	51
295 Unit tests for pzmodel/rebuild	52
290 Unit tests for prinodel/resp	00 57
297 Unit tests for prinodel/save	97 50
290 Unit tests for primodel/setUpits	09 61
299 Unit tests for principal (set lumits	01
201 Unit tests for primodel/set/Name	03
301 Unit tests for prinodel/set/Units. 9 202 Unit tests for prinodel/set/Units. 9	00
302 Unit tests for pzmodel/simplify	07



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

303 Unit tests for pzmodel/string	38
304 Unit tests for pzmodel/times	70
305 Unit tests for pzmodel/tomfir	2
306 Unit tests for pzmodel/tomiir	6
307 Unit tests for pzmodel/type	7
308 Unit tests for rational/char	78
309 Unit tests for rational/copy	79
310 Unit tests for rational/created	31
311 Unit tests for rational/creator	33
312 Unit tests for rational/display	34
313 Unit tests for rational/eq	36
314 Unit tests for rational/get	37
315 Unit tests for rational/getlowerFreq	38
316 Unit tests for rational/getupperFreq	39
317 Unit tests for rational/index.)1
318 Unit tests for rational/isprop)3
319 Unit tests for rational/loadobi.)4
320 Unit tests for rational/ne)5
321 Unit tests for rational/rational 90	99
322 Unit tests for rational/rebuild	000
323 Unit tests for rational/resp.	000
324 Unit tests for rational/save	05
325 Unit tests for rational/saturits	007
326 Unit tests for rational/setName	
320 Unit tests for rational/setOunits	111
327 Unit tests for rational/setOunts.)12
320 Unit tests for rational/type	112
329 Unit tests for rational/type)15)15
221 Unit tests for smodel/addRatess	110
222 Unit tests for smodel/deanAlioses	10
222 Unit tests for smodel/creat Anases)2U)91
224 Unit tests for smodel/copy	121 100
354 Unit tests for smodel/loadobj. 10 227 Unit tests for smodel/loadobj. 10	122
335 Unit tests for smodel/setAnases	124 197
227 Unit tests for smodel/setParameters.	121
337 Unit tests for smodel/setParams. 10 220 Unit tests for smodel/setParams. 10	130
338 Unit tests for smodel/set frans. 10 220 Unit tests for smodel/set frans. 10	132
339 Unit tests for smodel/set values	134
340 Unit tests for smodel/setAunits.	130
341 Unit tests for smodel/setAvals. 10^{-10}	138
342 Unit tests for smodel/setXvar	140
343 Unit tests for smodel/set Yunits.	142
344 Unit tests for smodel/smodel	145
345 Unit tests for ssm/addParameters	146
346 Unit tests for ssm/copy	147
347 Unit tests for ssm/loadobj	148
348 Unit tests for model/DFACS	149
349 Unit tests for model/IFO	150
350 Unit tests for ssm/ssm	153
351 Unit tests for time/datenum	154
352 Unit tests for time/double	155
353 Unit tests for time/format	156
354 Unit tests for time/minus	157



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

356 Unit tests for time/plus. 1 357 Unit tests for time/string. 1 358 Unit tests for time/time. 1 359 Unit tests for time/timeformat. 1	059 060 062 063 064 065 066
357 Unit tests for time/string. 1 358 Unit tests for time/time. 1 359 Unit tests for time/timeformat. 1	060 062 063 064 065 066
358 Unit tests for time/time. 1 359 Unit tests for time/timeformat. 1	062 063 064 065 066
359 Unit tests for time/timeformat.	063 064 065 066
	064 065 066
360 Unit tests for time/timezone	$065 \\ 066 \\ 068$
361 Unit tests for timespan/char	066 068
362 Unit tests for timespan/copy	068
363 Unit tests for timespan/created	000
364 Unit tests for timespan/creator	070
365 Unit tests for timespan/display	071
366 Unit tests for timespan/eq	073
367 Unit tests for timespan/get	074
368 Unit tests for timespan/index	076
369 Unit tests for timespan/isprop	078
370 Unit tests for timespan/loadobj	079
371 Unit tests for timespan/ne	080
372 Unit tests for timespan/rebuild	081
373 Unit tests for timespan/save	083
374 Unit tests for timespan/setEndT	085
375 Unit tests for timespan/setName	087
376 Unit tests for timespan/setStartT 1	089
377 Unit tests for timespan/string	090
378 Unit tests for timespan/timespan	093
379 Unit tests for timespan/type	094



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



1 Introduction

This document captures the results of the unit test run on LTPDA. The development cycle of LTPDA uses an automated test-rig which runs every night using a fresh build of the toolbox. The test-rig runs a series of different tests, including the unit tests presented in this report.

The LTPDA unit tests are gathered according to the object class and method within that class. For each unit test, there is a syntax test (to check that the method can be run) and an algorithm test (to check that the method does the correct thing).

The test tables presented in this document are structured as follows:

- Column 1 contains the test name or number, together with any sub-name or number
- Column 2 contains the general description of the test.
- Column 3, row 1 contains the description of the syntax test.
- Column 3, row 2 contains the description of the algorithm test.
- Column 4, row 1 contains the result of the syntax test (pass or fail).
- Column 4, row 2 contains the result of the algorithm test (pass or fail).

1.1 Version

This report refers to version 2.4 of the LTPDA toolbox.

2 Results

ao/abs			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the [ao/abs] method	Test that the [ao/abs] method	pass
02	works with a vector of objects	works for a vector of objects as	
	as input.	input.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/abs			
		1) Check that the number of el- ements in 'out' is the same as in 'vec' 2) Check that each output object contains the correct data.	pass
03	Tests that the [ao/abs] method works with a matrix of objects as input.	Test that the [ao/abs] method works for a matrix of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
04	Tests that the [ao/abs] method works with a list of objects as input.	Test that the [ao/abs] method works for a list of objects as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
05	Tests that the [ao/abs] method works with a mix of different arrays of objects as input.	Tests that the [ao/abs] method works with a mix of different ar- rays of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
06	Tests that the [ao/abs] method properly applies history.	Test that the result of applying the [ao/abs] method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[ao/abs]'. 2) Check that the re-built object is the same object as the input.	pass
07	Tests that the [ao/abs] method can modify the input AO.	Test that the [ao/abs] method can modify the input object by calling with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass
		1) Check that 'out' and 'aeq' are now different. 2) Check that 'aeq' is not changed 3) Check that the modified input is the [ao/abs] value of the copy 4) Check that out and amodi are the same	pass
08	Test that the [ao/abs] method uses the plist to get the axis.	Test that the [ao/abs] method uses the plist to get the axis.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/abs			
		1) Check that the [ao/abs] method applies to the x-axis 2) Check that the [ao/abs] method applies to the y-axis 3) Check that the [ao/abs] method applies to both axes 4) Check that the re-built object is the same as in 'out[13]'.	pass
09	Test the shape of the data in AOs.	Test that the [ao/abs] method keeps the data shape of the in- put object. The input AO data must be an array with row data and/or column data.	pass
		1) Check that the shape of the data doesn't change.	pass
10	Check that the [ao/abs] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
11	Check that the [ao/abs] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionally check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the same plotinfo plist	pass

Table 1: Unit tests for ao/abs.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/acos			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists	pass
		nave the correct parameters.	
02	Tests that the [ao/acos] method works with a vector of objects as input.	vorks for a vector of objects as input.	pass
		1) Check that the number of el-	pass
		'vec' 2) Check that each output	
	Tests that the [as /asss] method	Test that the lee lees method	Daga
03	works with a matrix of objects	works for a matrix of objects as	pass
	as input.	input.	
	I	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
04	Tests that the [ao/acos] method	Test that the [ao/acos] method	pass
04	works with a list of objects as	works for a list of objects as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/acos] method	Tests that the [ao/acos] method	pass
	works with a mix of different	works with a mix of different ar-	
	arrays of objects as input.	1) Check that the number of al	pagg
		ements in 'out' is the same as in	pass
		'mat' 2) Check that each output	
		object contains the correct data.	
0.0	Tests that the [ao/acos] method	Test that the result of applying	pass
06	properly applies history.	the [ao/acos] method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to ' $[ao/acos]$ '. 2) Check that the	
		re-built object is the same object	
		as the input.	
07	Tests that the [ao/acos] method	I lest that the [ao/acos] method	pass
	can modify the input AO.	can modify the input object by	
		the method decen't sharper the	
		input of the function notation	
		(with a equal sign)	
		((winn a chnar gign).	1



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/acos			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the [ao/acos]	
		value of the copy 4) Check that	
		out and amodi are the same	
	Test that the [ao/acos] method	Test that the [ao/acos] method	pass
08	uses the plist to get the axis.	uses the plist to get the axis.	1
		1) Check that the [ao/acos]	pass
		method applies to the x-axis 2)	1
		Check that the [ao/acos] method	
		applies to the v-axis 3) Check	
		that the [ao/acos] method ap-	
		plies to both axes 4) Check that	
		the re-built object is the same as	
		in 'out[13]'.	
	Test the shape of the data in	Test that the [ao/acos] method	pass
09	AOs.	keeps the data shape of the input	1
		object. The input AO data must	
		be an array with row data and/or	
		column data.	
		1) Check that the shape of the	pass
		data doesn't change.	pass
	Check that the [ao/acos]	Call the method with a list of	pass
10	method pass back the output	output variables and with a sin-	1
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	1
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/acos]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	1
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	*

Table 2: Unit tests for ao/acos.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/and			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
rule1 (tsdata and tsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and fsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/and			
rule1 (cdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (tsdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (tsdata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (cdata and tsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
tsdata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rulo1 (edata and	Tacts the arithmetic operators	Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
fsdata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and fsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule1 (cdata and	Tests the arithmetic operators	Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
xydata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single tsdata and vector tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector tsdata and single tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule2 (single	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
xydata and vector tsdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector tsdata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single cdata and vector tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
mule2 (verter todate	Tasts the arithmetic encretors	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and single cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single fsdata and vector fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector fsdata and single fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rulo2 (single	Tasts the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata and vector fsdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector fsdata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single cdata and vector fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule2 (vector fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and single cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	I ma
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single xydata and vector xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector xydata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule2 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector xydata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector xydata and single cdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector cdata and single cdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule2 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule3 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector tsdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule3 (vector fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector xydata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule3 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector fsdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	1
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector cdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector cdata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector cdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule3 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/and			
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule4 (vector fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector fsdata)	rule 4.	rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector fsdata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector fsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector xydata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector xydata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule4 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 4.	rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP tsdata and single tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single tsdata and NxP tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule5 (NxP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and single xydata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP tsdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule5 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	разы
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP fsdata and single fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single fsdata and NxP fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule5 (NxP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and single xydata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP fsdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule5 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP xydata and single xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule5 (NxP xydata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and single cdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single cdata and NxP xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP cdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule5 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP cdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rulo6 (NyP todata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule6 (NxP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector xydata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule6 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP fsdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	-
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector cdata)	rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule6 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP cdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule7 (NxP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule7 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP tsdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	-
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule7 (NxP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	Pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule7 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP fsdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule7 (NxP xydata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rulo7 (voctor edata	Tosts the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP cdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rules (NyP tsdata	Tasts the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule8 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP tsdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rulos (NyP fsdata	Tasts the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule8 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule8 (NyP yvdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector cdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	разы
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule8 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP cdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP tsdata and NxQ tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ tsdata and NxP tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP tsdata and NxQ xydata)	rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP tsdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule9 (NxQ cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP tsdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP fsdata and NxQ fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ fsdata and NxP fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
mile0 (NaD fodete	Trate the orithmetic energy and	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxQ xydata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP fsdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule9 (NxQ cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP xydata and NxQ xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
mule0 (NuD undete	Tasts the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxQ cdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ cdata and NxP xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP cdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule9 (NxQ cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP cdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rule10 (NyP tsdata	Tasts the arithmetic operators	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP xydata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the anthmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/and			
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and			
rula10 (NyP edata	Tests the arithmetic operators	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/and			
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/and			
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and			pass
tosta (fadoto ond	Tests all arithmatic aparations	Trata all arithmatic anonationa	pass
tsdata)	which are not allowed.	which are not allowed.	pass
		Here we test some negative cases.	pass
tests (tsdata and	Tests all arithmetic operations	Tests all arithmetic operations	pass
isdata)	which are not allowed.	which are not allowed.	
tasta (AO no data	Tests all arithmatic anonations	Here we test some negative cases.	pass
and tsdata)	which are not allowed	which are not allowed	pass
	which are not anowed.	Here we test some negative cases	nass
tests (tsdata and	Tests all arithmetic operations	Tests all arithmetic operations	pass
AO no data)	which are not allowed.	which are not allowed.	Pass
		Here we test some negative cases.	pass
tests (different fs in	Tests all arithmetic operations	Tests all arithmetic operations	pass
tsdata)	which are not allowed.	which are not allowed.	T
		Here we test some negative cases.	pass
tests (different x	Tests all arithmetic operations	Tests all arithmetic operations	pass
values in fsdata)	which are not allowed.	which are not allowed.	.
,		Here we test some negative cases.	pass
tests (negative	Tests all arithmetic operations	Tests all arithmetic operations	pass
test)	which are not allowed.	which are not allowed.	
		Here we test some negative cases.	pass
tests (negative	Tests all arithmetic operations	Tests all arithmetic operations	pass
test)	which are not allowed.	which are not allowed.	
		Here we test some negative cases.	pass


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/and

Table 3: Unit tests for ao/and.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/angle			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually	pass
		1) Check that got Info call ro	neg
		turned an minfe object in all	pass
		agos 2) Check that all plists	
		have the correct parameters	
	Tests that the [as /angle]	Test that the [ac/angle] method	naga
02	method works with a vector of	works for a water of objects as	pass
	chicata as input	input	
	objects as input.	1) Chock that the number of el	nase
		amonts in 'out' is the same as in	pass
		'voc' 2) Chock that each output	
		object contains the correct data	
	Tosts that the [ac/angle]	Tost that the [20/2nglo] method	nase
03	mothod works with a matrix of	works for a matrix of objects as	pass
	objects as input	input	
	objects as input.	1) Check that the number of el-	nase
		ements in 'out' is the same as in	pass
		'mat' 2) Check that each output	
		object contains the correct data	
	Tests that the [ao/angle]	Test that the [ao/angle] method	nase
04	method works with a list of	works for a list of objects as in-	pass
	objects as input	but	
	objects as input.	1) Check that the number of el-	nass
		ements in 'out' is the same as in	pass
		'mat' 2) Check that each output	
		object contains the correct data	
	Tests that the [ao/angle]	Tests that the [ao/angle] method	pass
05	method works with a mix of	works with a mix of different ar-	Pass
	different arrays of objects as	rays of objects as input	
	input.	1) Check that the number of el-	pass
	input.	ements in 'out' is the same as in	Pass
		'mat' 2) Check that each output	
		object contains the correct data.	
	Tests that the [ao/angle]	Test that the result of apply-	pass
06	method properly applies history.	ing the [ao/angle] method can be	P
		processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[ao/angle]'. 2) Check that the	
		re-built object is the same object	
		as the input.	
07	Tests that the [ao/angle]	Test that the [ao/angle] method	pass
07	method can modify the input	can modify the input object by	-
	AO.	calling with no output and that	
		the method doesn't change the	
		input of the function notation	
		(with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/angle			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the [ao/angle]	
		value of the copy 4) Check that	
		out and amodi are the same	
0.0	Test that the [ao/angle] method	Test that the [ao/angle] method	pass
08	uses the plist to get the axis.	uses the plist to get the axis.	
		1) Check that the [ao/angle]	pass
		method applies to the x-axis	
		2) Check that the [ao/angle]	
		method applies to the y-axis	
		3) Check that the [ao/angle]	
		method applies to both axes 4)	
		Check that the re-built object is	
		the same as in ' $out[13]$ '.	
00	Test the shape of the data in	Test that the [ao/angle] method	pass
09	AOs.	keeps the data shape of the input	
		object. The input AO data must	
		be an array with row data and/or	
		column data.	
		1) Check that the shape of the	pass
		data doesn't change.	
10	Check that the [ao/angle]	Call the method with a list of	pass
10	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/angle]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 4: Unit tests for ao/angle.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/ao			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists	pass
		have the correct parameters.	
02	Tests that the [ao/ao] method works with a vector of objects as input.	Test that the [ao/ao] method works for a vector of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'vec' 2) Check that each output object contains the correct data.	pass
03	Tests that the [ao/ao] method works with a matrix of objects as input.	Test that the [ao/ao] method works for a matrix of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
04	Tests that the [ao/ao] method works with a list of objects as input.	Test that the [ao/ao] method works for a list of objects as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
05	Tests that the [ao/ao] method works with a mix of different arrays of objects as input.	Tests that the [ao/ao] method works with a mix of different ar- rays of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
06	Tests that the [ao/ao] method properly applies history.	Test that the result of applying the [ao/ao] method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[ao/ao]'. 2) Check that the re- built object is the same object as the input.	pass
08	Tests that the ao method properly applies history to the	Test that the output can be pro- cessed back to an m-file.	pass
	ASCII-file constructor.	1) Check that the last entry in the history of 'out' corresponds to 'ao'. 2) Check that the rebuilt objects are the same as 'out1' and 'ouit2'	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/ao			
00	Tests that the ao method	Test that the output can be pro-	pass
03	properly applies history to the	cessed back to an m-file.	
	complex ASCII-file constructor.	1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'ao'. 2) Check that the read	
		data are correct 3) Check that	
		the rebuilt object is the same as	
		in 'out15'	
10	Tests that the ao method	Test that the output can be pro-	pass
13	properly applies history to the	cessed back to an m-file.	-
	vals constructor.	1) Check that the last entry in	pass
		the history of 'out' corresponds	1
		to 'ao'. 2) Check that the rebuilt	
		object is the same as 'out'.	
	Tests that the ao method	Test that the output can be pro-	pass
15	properly applies history to the	cessed back to an m-file.	1
	plist(fcn) constructor.	1) Check that the last entry in	pass
	r st(t) it statistic	the history of 'out' corresponds	T
		to 'ao'. 2) Check that the rebuilt	
		object is the same as 'out'.	
	Tests that the ao method	Test that the output can be pro-	pass
16	properly applies history to the	cessed back to an m-file.	P
	plist(vals) constructor.	1) Check that the last entry in	pass
		the history of 'out' corresponds	Pass
		to 'ao'. 2) Check that the rebuilt	
		object is the same as 'out'.	
	Tests that the ao method	Test that the output can be pro-	pass
17	properly applies history to the	cessed back to an m-file.	T
	plist(x/v-vals) constructor.	1) Check that the last entry in	pass
	r (/ J · · · ·) · · · · · · · · ·	the history of 'out' corresponds	T
		to 'ao'. 2) Check that the rebuilt	
		object is the same as 'out'.	
10	Tests that the ao method	Test that the output can be pro-	pass
18	properly applies history to the	cessed back to an m-file.	-
	plist(tsfcn) constructor.	1) Check that the last entry in	pass
		the history of 'out' corresponds	-
		to 'ao'. 2) Check that the rebuilt	
		object is the same as 'out'.	
10	Tests that the ao method	Test that the output can be pro-	pass
19	properly applies history to the	cessed back to an m-file.	
	plist(fsfcn) constructor.	1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'ao'. 2) Check that the rebuilt	
		objects are the same as 'out1'	
		and 'out2'	
00	Tests that the ao method	Test that the output can be pro-	pass
20	properly applies history to the	cessed back to an m-file.	
I	plist(win) constructor.	L	1



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/ao			
		1) Check that the last entry in the history of 'out' corresponds to 'ao'. 2) Check that the rebuilt object is the same as 'out'.	pass
21	Tests that the ao method properly applies history to the	Test that the output can be processed back to an m-file.	pass
	plist(waveform) constructor.	1) Check that the last entry in the history of 'out' corresponds to 'ao'. 2) Check that the rebuilt objects are the same as 'out'.	pass
23	Tests that the ao method properly applies history to the	Test that the output can be pro- cessed back to an m-file.	pass
	plist(polynomial) constructor.	1) Check that the last entry in the history of 'out' corresponds to 'ao'. 2) Check that the rebuilt object is the same as 'out'.	pass
25	Tests that the ao method properly applies history to the	Test that the output can be processed back to an m-file.	pass
	plist(pzmodel) constructor.	1) Check that the last entry in the history of 'out' corresponds to 'ao'. 2) Check that the rebuilt object is the same as 'out'.	pass
26	Tests that the ao method properly applies history to the	Test that the output can be pro- cessed back to an m-file.	pass
	data-object constructor.	1) Check that the last entry in the history of 'out' corresponds to 'ao'.	pass
28	Tests that the ao method properly applies history to the	Test that the output can be pro- cessed back to an m-file.	pass
	x-vector, y-vector constructor.	1) Check that the last entry in the history of 'out' corresponds to 'ao'. 2) Check that the rebuilt object is the same as 'out'.	pass
29	Tests that the ao method properly applies history to the	Test that the output can be pro- cessed back to an m-file.	pass
	filename + plist constructor.	1) Check that the last entry in the history of 'out' corresponds to 'ao'. 2) Check that the rebuilt object is the same as 'out'.	pass
30	Tests that the ao method properly applies history to the	Test that the output can be pro- cessed back to an m-file.	pass
	filename + plist constructor.	 Check that the last entry in the history of 'out' corresponds to 'ao'. 2) Check the read data Check that the rebuilt object is the same as 'out' 	pass
31	Tests that the ao method properly applies history to the filename + plist constructor	Test that the output can be processed back to an m-file.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/ao			
		1) Check that the last entry in the history of 'out' corresponds to 'ao'. 2) Check that the rebuilt object is the same as 'out'.	pass
60	Tests that the constructor method doesn't apply history to the read MAT-file constructor.	Tests that the constructor method doesn't apply history to the read MAT-file constructor.	pass
		1) Check that the history is the same as the history of the saved object. Because save and load shouldn't add a history step. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
61	Tests that the constructor properly applies history to the read XML-file constructor.	Tests that the constructor prop- erly applies history to the read XML-file constructor.	pass
		1) Check that the history is the same as the history of the saved object. Because save and load shouldn't add a history step. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
62	Tests that the constructor properly applies history in the struct constructor.	Tests that the constructor prop- erly applies history in the struct constructor.	pass
		1) Check that the last entry in the history of 'out' corresponds to the class name. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
64	Tests that the contructor properly applies history to the plist(filename) constructor.	Tests that the contructor prop- erly applies history to the plist(filename) constructor.	pass
		1) Check that the save method doesn't change the input ob- ject 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
65	Tests that the contructed object can be submitted and retrieved.	Tests that the contructed objectcan be submitted and retrieved.1) Check that the last entry in	pass pass
		the history of 'out' corresponds to the class name. 2) Check that the 'rebuild' method produces the same object as 'out'.	r
66	Tests that the constructor	Tests that the constructor	pass
	properly works with the	properly works with the	
	plist(pzmodel) constructor.	plist(pzmodel) constructor.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/ao			
		1) Check that the last entry in the history of 'out' corresponds to 'ssm'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
67	Tests that the constructor properly applies history to the pole/zero model + plist object constructor.	Tests that the constructor properly applies history to the pole/zero model + plist object constructor.	pass
		1) Check that the last entry in the history of 'out' corresponds to class name. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
68	Tests that the constructor properly applies history to the conn+Id constructor.	Tests that the constructor properly applies history to the conn+Id constructor.	pass
		1) Check that the last entry in the history of 'out' corresponds to class name. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass

Table 5: Unit tests for ao/ao.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/asin			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the [ao/asin] method	Test that the [ao/asin] method	pass
02	works with a vector of objects	works for a vector of objects as	
	as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'vec' 2) Check that each output	
		object contains the correct data.	
03	Tests that the [ao/asin] method	Test that the [ao/asin] method	pass
	works with a matrix of objects	works for a matrix of objects as	
	as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
04	Tests that the [ao/asin] method	Test that the [ao/asin] method	pass
01	works with a list of objects as	works for a list of objects as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/asin] method	Tests that the [ao/asin] method	pass
	works with a mix of different	works with a mix of different ar-	
	arrays of objects as input.	rays of objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		(mat' 2) Check that each output	
		object contains the correct data.	
06	Tests that the [ao/asin] method	Test that the result of applying	pass
	properly applies history.	the [ao/asin] method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to $[ao/asin]^2$. 2) Check that the	
		re-built object is the same object	
		as the input.	
07	Tests that the [ao/asin] method	I lest that the [ao/asin] method	pass
	can modify the input AO.	can modify the input object by	
		calling with no output and that	
		the method doesn't change the	
		input of the function notation	
		(with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/asin			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the [ao/asin]	
		value of the copy 4) Check that	
		out and amodi are the same	
	Test that the [ao/asin] method	Test that the [ao/asin] method	pass
08	uses the plist to get the axis.	uses the plist to get the axis.	1
		1) Check that the [ao/asin]	pass
		method applies to the x-axis 2	1
		Check that the [ao/asin] method	
		applies to the y-axis 3) Check	
		that the [ao/asin] method applies	
		to both axes 4) Check that the	
		re-built object is the same as in	
		out[13]'.	
	Test the shape of the data in	Test that the [ao/asin] method	pass
09	AOs.	keeps the data shape of the in-	P
		put object. The input AO data	
		must be an array with row data	
		and/or column data	
		1) Check that the shape of the	pass
		data doesn't change	Pass
	Check that the [ao/asin] method	Call the method with a list of	pass
10	pass back the output objects to	output variables and with a sin-	Pass
	a list of output variables or to a	gle output variable Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	P
		iects 2) Check that the 'rebuild'	
		method produces the same ob-	
		iect as 'out'.	
	Check that the [ao/asin] method	Call the method with a list of	pass
11	pass back the output objects to	output variables and with a sin-	r
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	r and

Table 6: Unit tests for ao/asin.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/atan			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minio object in all	
		cases. 2) Check that all plists	
		The the correct parameters.	
02	rests that the [ao/atan] method	rest that the [ao/atan] method	pass
	works with a vector of objects	works for a vector of objects as	
	as input.	1) Chock that the number of al	page
		amonts in 'out' is the same as in	pass
		'voc' 2) Chock that each output	
		object contains the correct data	
	Tests that the [ao/atan] method	Test that the [ao/atan] method	nase
03	works with a matrix of objects	works for a matrix of objects as	pass
	as input	input	
	as input.	1) Check that the number of el-	nass
		ements in 'out' is the same as in	pass
		'mat' 2) Check that each output	
		object contains the correct data	
	Tests that the [ao/atan] method	Test that the [ao/atan] method	pass
04	works with a list of objects as	works for a list of objects as in-	pass
	input.	put.	
	1	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/atan] method	Tests that the [ao/atan] method	pass
60	works with a mix of different	works with a mix of different ar-	
	arrays of objects as input.	rays of objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
06	Tests that the [ao/atan] method	Test that the result of applying	pass
	properly applies history.	the [ao/atan] method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[ao/atan]'. 2) Check that the	
		re-built object is the same object	
		as the input.	
07	Tests that the [ao/atan] method	Test that the [ao/atan] method	pass
	can modify the input AO.	can modify the input object by	
		calling with no output and that	
		the method doesn't change the	
		input of the function notation	
		(with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/atan			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the [ao/atan]	
		value of the copy 4) Check that	
		out and amodi are the same	
	Test that the [ao/atan] method	Test that the [ao/atan] method	pass
08	uses the plist to get the axis.	uses the plist to get the axis.	-
		1) Check that the [ao/atan]	pass
		method applies to the x-axis 2)	1
		Check that the [ao/atan] method	
		applies to the y-axis 3) Check	
		that the [ao/atan] method ap-	
		plies to both axes 4) Check that	
		the re-built object is the same as	
		in 'out[13]'.	
	Test the shape of the data in	Test that the [ao/atan] method	pass
09	AOs.	keeps the data shape of the input	1
		object. The input AO data must	
		be an array with row data and/or	
		column data	
		1) Check that the shape of the	pass
		data doesn't change.	Pass
	Check that the [ao/atan]	Call the method with a list of	pass
10	method pass back the output	output variables and with a sin-	P
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	P
		iects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/atan]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionalv	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	·

Table 7: Unit tests for ao/atan.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/atan2			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the atan2 method	Tests that the atan2 method	pass
02	works only with two AOs as	works only with two AOs as in-	
	input	put	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'atvec' 2) Check that each output	
		AO contains the correct data.	
03	Tests that the atan2 method	Test that the result of applying	pass
00	properly applies history.	the atan2 method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'atan2'. 2) Check that the re-	
		built object is the same object as	
		the input.	
04	Tests that the atan2 method	Tests that the atan2 method	pass
04	can not be used as a modifier	can not be used as a modifier	
	method.	method. The command should	
		fail.	
		1) Nothing to test.	pass

Table 8: Unit tests for ao/atan2.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/average			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the average method works with a vector of AOs as input.	Tests that the average method works with a vector of AOs as in- put.	pass
		1) Check that the outputs have exactly one AO 2) Check that the outputs have the correct data. 3) Check the rebuilt objects	pass
03	Tests that the average method works with a matrix of AOs as input.	Tests that the average method works with a matrix of AOs as input.	pass
		1) Check that the outputs have exactly one AO 2) Check that the outputs have the correct data. 3) Check the rebuilt objects	pass
04	Tests that the average method works with a list of AOs as input.	Tests that the average method works with a list of AOs as in- put.	pass
		1) Check that the outputs have exactly one AO 2) Check that the outputs have the correct data. 3) Check the rebuilt objects	pass
05	Tests that the average method works with a mix of different shaped AOs as input.	Tests that the average method works with a mix of different shaped AOs as input.	pass
		1) Check that the output is exact one AO 2) Check that the output have the correct data.	pass
06	Tests that the [ao/average] method properly applies history.	Test that the result of applying the [ao/average] method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[ao/average]'. 2) Check that the re-built object is the same object as the input.	pass
07	Tests that the average method cannot modify the input AO. The method must throw an error for the modifier call	Test that the average method cannot modify the input AO by calling with no output	pass
	i or or one mounter can.	I T TOUTING TO CHOOK.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/average			
08	Test the shape of the output	Test that the average method	pass
00	Test the shape of the output.	keeps the data shape of the in-	
		put object. The input AO must	
		be an AO with row data and an	
		AO with column data.	
		1) Check that the shape of the	pass
		data doesn't change.	
10	Tests that the average method	Tests that the average method	pass
10	works with a single AO as input.	works with a single AO as input.	
		1) Check that the outputs have	pass
		exactly one AO 2) Check that the	
		outputs have the correct data. 3)	
		Check the rebuilt objects	

Table 9: Unit tests for ao/average.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



bin/data			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists	pass
		have the correct parameters.	
02	Tests that the [bin/data] method works with a vector of objects as input.	Test that the [bin/data] method works for a vector of objects as input.	pass
	r J. T. F. T.	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'vec' 2) Check that each output	
	Trasta that the [bin /data]	Object contains the correct data.	
03	method works with a matrix of	works for a matrix of objects as	pass
	objects as input	input	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	pass
		'mat' 2) Check that each output	
		object contains the correct data.	
04	Tests that the [bin/data]	Test that the [bin/data] method	pass
04	method works with a list of	works for a list of objects as in-	
	objects as input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	<u> </u>
05	Tests that the [bin/data]	Tests that the [bin/data] method	pass
	method works with a mix of	works with a mix of different ar-	
	different arrays of objects as	1) Check that the number of al	pagg
	mput.	amonts in 'out' is the same as in	pass
		'mat' 2) Check that each output	
		object contains the correct data.	
	Tests that the [bin/data]	Test that the result of apply-	pass
06	method properly applies history.	ing the [bin/data] method can be	1
		processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[bin/data]'. 2) Check that the	
		re-built object is the same object	
		as the input.	<u> </u>
07	1 lests that the [bin/data]	1 Test that the [bin/data] method	pass
	method can modify the input	can modify the input object by	
	AU.	the method decar't sharper the	
		input of the function notation	
		(with a equal sign)	
1		(minin a oquar sign).	1



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

bin/data			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the [bin/data]	
		value of the copy 4) Check that	
		out and amodi are the same	
	Check that the [bin/data]	Call the method with a list of	pass
10	method pass back the output	output variables and with a sin-	Pass
	objects to a list of output	gle output variable Additionaly	
	variables or to a single variable	check that the rebuild method	
	variables of to a single variable.	works on the output	
		1) Check that the output con	nage
		toing the right number of ob	pass
		isota 2) Chook that the 'robuild'	
		method produces the same of	
		inethod produces the same ob-	
		ject as out.	
11	Check that the [bin/data]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	
17	Tests handling of units: 1)	1) Rebinning the data	pass
	rebinning of the PSD data 2)	1) Check that (rebinned PSD yu-	pass
	rebinning of the LPSD data 3)	nits) equals (input PSD yunits)	
	compares the units of the input	2) Check that (rebinned PSD xu-	
	and output	nits) equals (input PSD xunits)	
		3) Check that (rebinned LPSD	
		yunits) equals (input LPSD yu-	
		nits) 4) Check that (rebinned	
		LPSD xunits) equals (input	
		LPSD xunits)	
10	Tests handling of units: 1)	1) Rebinning the data	pass
10	rebinning of the PSD data 2)	1) Check that (rebinned PSD yu-	pass
	rebinning of the LPSD data 3)	nits) equals (input PSD yunits)	
	compares the units of the input	2) Check that (rebinned PSD xu-	
	and output	nits) equals (input PSD xunits)	
	_	3) Check that (rebinned LPSD	
		yunits) equals (input LPSD vu-	
		nits) 4) Check that (rebinned	
		LPSD xunits) equals (input	
		LPSD xunits)	
	Tests handling of units: 1)	1) Rebinning the data	pass
19	rebinning of the PSD data 2)		1
	rebinning of the LPSD data 3)		
	compares the units of the input		
	i parte the second of the input		

and output



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

bin/data			
		 Check that (rebinned PSD yunits) equals (input PSD yunits) Check that (rebinned PSD xunits) equals (input PSD xunits) Check that (rebinned LPSD yunits) equals (input LPSD yunits) equals (input LPSD yunits) Check that (rebinned LPSD yunits) equals (input LPSD yunits) 	pass
		LPSD xunits) equals (input LPSD xunits)	
20	Tests handling of units: 1)	1) Rebinning the data	pass
	rebinning of the PSD data 2)	Nothing to check	pass
91	Telsiusning.dlint.geofLiPSES:da)ta	1) Rebinning the data	pass
21	rebinning of the PSD data 2)	Nothing to check	pass

rebinning of the LPSD data

Table 10: Unit tests for bin/data.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/cat			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
00	Tests that the [ao/cat] method	Test that the [ao/cat] method	pass
02	works with a vector of objects	works for a vector of objects as	
	as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	-
		'vec' 2) Check that each output	
		object contains the correct data.	
	Tests that the [ao/cat] method	Test that the [ao/cat] method	pass
03	works with a matrix of objects	works for a matrix of objects as	-
	as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	-
		'mat' 2) Check that each output	
		object contains the correct data.	
0.4	Tests that the [ao/cat] method	Test that the [ao/cat] method	pass
04	works with a list of objects as	works for a list of objects as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/cat] method	Tests that the [ao/cat] method	pass
05	works with a mix of different	works with a mix of different ar-	
	arrays of objects as input.	rays of objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
0.0	The cat method doesn't apply	The cat method doesn't apply	pass
00	history.	history.	
		1) Nothing to test	pass
07	The cat method can not be used	The cat method can not be used	pass
07	as a modifer method.	as a modifer method. In this case	-
		throws the method an error.	
		1) Nothing to test.	pass

Table 11: Unit tests for ao/cat.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/char			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the char method	Test that the char method works	pass
02	works with a vector of AOs as	for a vector of AOs as input.	
	input.	1) Check that the output contain	pass
		at least each object name	
0.2	Tests that the char method	Test that the char method works	pass
05	works with a matrix of AOs as	for a matrix of AOs as input.	
	input.	1) Check that the output contain	pass
		at least each object name	
0.4	Tests that the char method	Test that the char method works	pass
04	works with a list of AOs as	for a list of AOs as input.	
	input.	1) Check that the output contain	pass
		at least each object name	
05	Tests that the char method	Test that the char method works	pass
05	works with a mix of different	with an input of matrices and	
	shaped AOs as input.	vectors and single AOs.	
		1) Check that the output contain	pass
		at least each object name	
06	Tests that the char method	The method char doesn't change	pass
00	properly applies history.	the data, thus it is not possible	
		to check the history. Nothing to	
		do.	
			pass

Table 12: Unit tests for a o/char.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/cohere			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually. 1) Check that getInfo call re-	pass pass
		turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	
02	Tests that the cohere method works with a vector of AOs as input.	Test that the cohere method works for a vector of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is 1 2) Check that each output AO contains the cor- rect data.	pass
03	Test that the cohere method doesn't work for a matrix of AOs as input.	Test that the cohere method doesn't work for a matrix of AOs as input.	pass
04	Tests that the cohere method works with a list of AOs as input (only two objects).	Test that the cohere method works for a list of AOs as input. (only two objects)	pass
		1) Check that the number of ele- ments in 'out' is 1. 2) Check that each output AO contains the cor- rect data.	pass
05	Test that the cohere method doesn't work with an input of matrices and vectors and single AOs.	Test that the cohere method doesn't work with an input of matrices and vectors and single AOs.	pass
06	Tests that the cohere method properly applies history.	1) Nothing to check Test that the result of applying the cohere method can be processed back to an m-file.	pass pass
		1) Check that the last entry in the history of 'out' corresponds to 'cohere'. 2) Check that the re- built object is the same as 'out'.	pass
07	Tests that the cohere method can not modify the input AO.	Test that the cohere method can not modify the input AO. The method must throw an error for the modifier call.	pass
08	Test the shape of the output.	Test that the plus method keeps the data shape of the input ob- ject. The input AO must be an AO with row data and an AO with column data.	pass pass
		1) Check that the shpe of the output data doesn't change.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/cohere			
09	Check that the cohere method	This test is not longer necessary	pass
	pass back the output objects to	because the cohere method pass	
	a list of output variables or to a	back always only one object.	
	single variable.	1) Nothing to check.	pass
10	Tests that the cohere method	Test that the applying cohere	pass
10	agrees with MATLAB's	works on two AOs.	
	mscohere when configured to	1) Check that output agrees with	pass
	use the same parameters.	the output of MATLAB's msco-	
		here. 2) Check that the shape of	
		the output data is equal to the	
		input data	
11	Check that the [ao/cohere]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	
12	Tests symmetry properties of	1) Prepare the test tsdata: white	pass
	complex-coherence: 1) white	noise from normal distribution +	
	noise produced from normal	offset 2) Assign a random unit	
	pdf, with a given mean value	3) Prepare the test tsdata: white	
	and sigma (distribution's 1st	noise from normal distribution +	
	and 2nd orders) 2) white noise	offset 4) Assign a random unit 5)	
	produced from normal pdf, with	complex coherence of the white	
	a given mean value and sigma	noise	
	(distribution's 1st and 2nd	1) Check that $C(x,y)$ equals	pass
	orders) 3) complex concrence of	C(y,x) 2) Check that $C(y,x)$ 2) Check that	
	the white hoise series 4)	C(x,x) equals 1 2) Check that $C(x,x)$ equals 1	
	Tent (C(mut)) 5th componenti C(mty)	1) Propage the test todate: white	naga
13	and deve a hour and a state of the state of	noise from normal distribution +	pass
	noise produced from normal	offset 2) Assign a random unit	
	noise produced from format	3) Prepare the test tsdata: white	
	and sigma (distribution's 1st	noise from normal distribution +	
	and 2nd orders) 2) white noise	offset 4) Assign a random unit 5)	
	produced from normal pdf, with	magnitude-squared coherence of	
	a given mean value and sigma	the white noise	
	(distribution's 1st and 2nd	1) Check that $C(x,y)$ equals	pass
	orders) 3) magnitude-squared	C(y,x) 1) Check that $C(x,x)$	1
	coherence of the white noise	equals 1 1) Check that $C(y,y)$	
	series 4) compare $C(x,y)$ with	equals 1	
14	Tests syncerpage of extiles and	1) Prepare the test tsdata: white	pass
14	C(my)ex-itoherence: 1) white	noise from normal distribution +	-
	noise produced from normal	offset 2) Assign a random unit	
	pdf, with a given mean value	3) Prepare the test tsdata: white	
	and sigma (distribution's 1st	noise from normal distribution +	
	and 2nd orders) 2) white noise	offset 4) Assign a random unit 5)	
	produced from normal pdf, with	complex coherence of the combi-	
	a given mean value and sigma	nation of noise	
	(distribution's 1st and 2nd		-
	orders) 3) complex coherence of		
	the combination of white noise		
	series 4) compare $C(x,y)$ with 1		94



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/cohere			
		1) Check that the complex coher-	pass
		ence equals 1	
15	Tests symmetry properties of	1) Prepare the test tsdata: white	pass
10	complex-coherence: 1) white	noise from normal distribution +	
	noise produced from normal	offset 2) Assign a random unit	
	pdf, with a given mean value	3) Prepare the test tsdata: white	
	and sigma (distribution's 1st	noise from normal distribution +	
	and 2nd orders) 2) white noise	offset 4) Assign a random unit 5)	
	produced from normal pdf, with	magnitude-squared coherence of	
	a given mean value and sigma	the combination of noise	
	(distribution's 1st and 2nd	1) Check that the magnitude-	pass
	orders) 3) magnitude-squared	squared coherence equals 1	
16	Testersycanole the prophinistic of	1) Prepare the test tsdata: white	pass
	whitplewisoheninge4)100whitee	noise from normal distribution +	
	Courses protthced from normal	offset 2) Assign a random unit	
	pdf, with a given mean value	3) Prepare the test tsdata: white	
	and sigma (distribution's 1st	noise from normal distribution +	
	and 2nd orders) 2) white noise	offset 4) Assign a random unit 5)	
	produced from normal pdf, with	magnitude-squared coherence of	
	a given mean value and sigma	the combination of noise	
	(distribution's 1st and 2nd	1) Check that the magnitude-	pass
	orders) 3) magnitude-squared	squared coherence equals the	
	coherence M of the combination	square modulus of the complex	
	of white noise series 4) complex	coherence	
17	Telses characed ing the initial aviante	1) Prepare the test tsdata: white	pass
	abischipeonloisedstring 5)ocmapare	noise from normal distribution +	
	pds(G)i2hvatgivan mean value	offset 2) Assign a random unit	
	and sigma (distribution's 1st	3) Prepare the test tsdata: white	
	and 2nd orders) 2) white noise	noise from normal distribution +	
	produced from normal pdf, with	offset 4) Assign a random unit 5)	
	a given mean value and sigma	complex cohere of the white noise	
	(distribution's 1st and 2nd	1) Check that (complex coher-	pass
	orders) 3) complex coherence of	ence yunits) equals [1] 2) Check	
	the white noise series 4)	that (complex coherence xunits)	
	compares the units of the input	equals [Hz]	
18	alestoutantling of units: 1) white	1) Prepare the test tsdata: white	pass
	noise produced from normal	noise from normal distribution +	
	pdf, with a given mean value	offset 2) Assign a random unit	
	and sigma (distribution's 1st	3) Prepare the test tsdata: white	
	and 2nd orders) 2) white noise	noise from normal distribution +	
	produced from normal pdf, with	offset 4) Assign a random unit 5)	
	a given mean value and sigma	magnitude-squared cohere of the	
	(distribution's 1st and 2nd	white noise	
	orders) 3) magnitude-squared	1) Check that (magnitude-	pass
	concernce of the white noise	squared coherence yunits) equals	
	series 4) compares the units of	[1] 2) Check that (magnitude-	
	the input and output	squared concernce xunits) equals	
10	Lests that differently sized data	Lest that applying cohere works	Dass
19		rest that applying cohere works	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/cohere			
		1) Check that cohere used the	pass
		length of the shortest ao.	
20	Tests that applying a single	Test that applying cohere works	pass
20	window the coherence is 1	on two AOs.	
		1) Check that the calculated co-	pass
		here is 1	
01	Tests the possibility to set the	1) Prepare the test tsdata: white	pass
21	number of averages rather than	noise from normal distribution +	
	setting the Nfft: 1) white noise	offset 2) cohere of the noise, with-	
	produced from normal pdf,	out detrending, random window,	
	with: a given mean value and	set number of averages	
	sigma (distribution's 1st and	1) Check that calculated navs are	pass
	2nd order) 2) cohere of the	identical to those requested	
202	Tests, thit possibility ndisgt the	1) white noise produced from	pass
	randbez afiaderagestrathebehah	uniform pdf, with: a given mean	
	sounageshie) Nhitcki the hiffectionse	value and sigma (distribution's	
	prombree of framer agais form pdf,	1st and 2nd order) 2) cohere	
	with: a given mean value and	of the noise, without detrending,	
	sigma (distribution's 1st and	random window, random navs 3)	
	2nd order) 2) cohere of the	get the number of averages 4) get	
	noise, without detrending,	the nfft used 5) run cohere again,	
	random window, random navs	with the nfft used	
	3) get the number of averages 4)	1) Check that calculated objects	pass
	get the nfft used 5) run cohere	C1 and C2 are identical	
23	agests, twetpossebility usede6) the	1) white noise produced from	pass
	nompercotheveasges tedheb jecas	normal pdf, with: a given mean	
	setting the Nfft: 1) white noise	value and sigma (distribution's	
	produced from normal pdf,	1st and 2nd order) 2) cohere	
	with: a given mean value and	of the noise, without detrending,	
	sigma (distribution's 1st and	random window, random navs 3)	
	2nd order) 2) cohere of the	get the number of averages 4) get	
	noise, without detrending,	the nfft used 5) run cohere again,	
	random window, random navs	with the nfft used 6) run cohere	
	3) get the number of averages 4)	again, with conflicting parame-	
	get the nfft used 5) run cohere	ters, and verify it uses nift rather	
	again, with the fift used 6)	than navs	
	compare navs, nfft, coheres	1) Check that calculated objects	pass
		C1 and $C2$ are identical 2) Check	
		that U3 used different values	
24	agrees with MATLAD's	works on two AOs	pass
	agrees with MAILAD's	1) Check that output arread with	negg
	use the same parameters	the output of MATLAR's mass	pass
	use the same parameters.	here 2) Check that the shape of	
		the output data is equal to the	
		input data	
		mput data	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/cohere			
05	Tests handling of units: 1) white	1) Prepare the test tsdata: white	pass
20	noise produced from normal	noise from normal distribution +	
	pdf, with a given mean value	offset 2) Assign a random unit	
	and sigma (distribution's 1st	3) Prepare the test tsdata: white	
	and 2nd orders) 2) white noise	noise from normal distribution +	
	produced from normal pdf, with	offset 4) Assign a random unit 5)	
	a given mean value and sigma	complex cohere of the white noise	
	(distribution's 1st and 2nd	1) Check that (complex coher-	pass
	orders) 3) complex coherence of	ence yunits) equals [1] 2) Check	
	the white noise series 4)	that (complex coherence xunits)	
	compares the units of the input	equals [Hz]	
0.0	Testsubputling of units: 1) white	1) Prepare the test tsdata: white	pass
20	noise produced from normal	noise from normal distribution +	
	pdf, with a given mean value	offset 2) Assign a random unit	
	and sigma (distribution's 1st	3) Prepare the test tsdata: white	
	and 2nd orders) 2) white noise	noise from normal distribution +	
	produced from normal pdf, with	offset 4) Assign a random unit 5)	
	a given mean value and sigma	complex cohere of the white noise	
	(distribution's 1st and 2nd	1) Check that (complex coher-	pass
	orders) 3) complex coherence of	ence yunits) equals [1] 2) Check	
	the white noise series 4)	that (complex coherence xunits)	
	compares the units of the input	equals [Hz]	
20	Testoutputling of special cases:	1) Prepare the test tsdata: white	pass
00	1) white noise produced from	noise from normal distribution +	
	normal pdf, with a given mean	offset 2) Assign a random unit 3)	
	value and sigma (distribution's	Prepare the test tsdata: the same	
	1st and 2nd orders) 2) the same	data as 1) and 2) 4) cohere of the	
	noise series 3) cohere of the	series	
	white noise series 4) compares	1) Check that calculated cohere	pass
	the output to unity	equals 1	

Table 13: Unit tests for ao/cohere.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/complex			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the complex method works with a vector of AOs as input.	Test that the complex method works for a vector of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is 1. 2) Check that the output AO contains the cor- rect data.	pass
03	Tests that the complex method works with a list of AOs as	Test that the complex method works for a list of AOs as input.	pass
	input.	1) Check that the number of ele- ments in 'out' is 1. 2) Check that the output AO contains the cor- rect data.	pass
05	Tests that the complex method can not modify the input AO.	Test that the complex method can not modify the input AO. The method must throw an er- ror for the modifier call.	pass
06	Tests that the [ao/complex] method properly applies history.	Test that the result of applying the [ao/complex] method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[ao/complex]'. 2) Check that the re-built object is the same object as the input.	pass
07	Control the method with a plist.	Test that the complex method keeps the data shape of the first input object. the input AO must be an AO with row data and an AO with column data.	pass
		1) Check that the shpe of the data doesn't change.	pass
11	Check that the [ao/complex] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the same plotinfo plist	pass

Table 14: Unit tests for ao/complex.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/compute			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the complex method works with a vector of AOs as input.	Test that the complex method works for a vector of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is 1. 2) Check that each output AO contains the cor- rect data.	pass
03	Tests that the compute method works with a matrix of AOs as input.	Test that the compute method works for a matrix of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is 2. 2) Check that each output AO contains the cor- rect data.	pass
04	Tests that the compute method works with a list of AOs as	Test that the compute method works for a list of AOs as input.	pass
	mpuo.	ments in 'out' is 2. 2) Check that each output AO contains the cor- rect data.	pass
05	Tests that the compute method works with a mix of different shaped AOs as input.	Test that the compute method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of ele- ments in 'out' is 2. 2) Check that each output AO contains the cor- rect data.	pass
06	Tests that the compute method applies no history.	Test that the result of applying the compute method can be pro- cessed back to an m-file.	pass
		1) Check that the last entry in the history of 'out' is not 'com- pute'. 2) Check that the rebuilt object is the same as 'out'.	pass
11	Check that the [ao/compute] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output	pass
		1) Check that the output con- tains the same plotinfo plist	pass

Table 15: Unit tests for ao/compute.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/conj			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minio object in all	
		cases. 2) Check that all plists	
	Trata that the [as /seri] method	The the correct parameters.	magg
02	rests that the [ao/conj] method	rest that the [ao/conj] method	pass
	works with a vector of objects	input	
	as input.	1) Chock that the number of al	neg
		amonts in 'out' is the same as in	pass
		'voc' 2) Chock that each output	
		object contains the correct data	
	Tests that the [ao/coni] method	Test that the [ao/coni] method	nase
03	works with a matrix of objects	works for a matrix of objects as	pass
	as input	input	
	as input.	1) Check that the number of el-	pass
		ements in 'out' is the same as in	Pass
		'mat' 2) Check that each output	
		object contains the correct data.	
	Tests that the [ao/coni] method	Test that the [ao/coni] method	pass
04	works with a list of objects as	works for a list of objects as in-	P
	input.	put.	
	1	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/conj] method	Tests that the [ao/conj] method	pass
05	works with a mix of different	works with a mix of different ar-	
	arrays of objects as input.	rays of objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
06	Tests that the [ao/conj] method	Test that the result of applying	pass
	properly applies history.	the [ao/conj] method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to $'[ao/conj]'$. 2) Check that the	
		re-built object is the same object	
		as the input.	
07	Tests that the [ao/conj] method	I lest that the [ao/conj] method	pass
	can modify the input AO.	can modify the input object by	
		calling with no output and that	
		the method doesn't change the	
		input of the function notation	
		(with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/conj			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the [ao/conj]	
		value of the copy 4) Check that	
		out and amodi are the same	
	Test that the [ao/coni] method	Test that the [ao/coni] method	pass
08	uses the plist to get the axis	uses the plist to get the axis	Pass
		1) Check that the [ao/coni]	nass
		method applies to the y-axis 2)	pass
		Check that the [so/coni] method	
		applies to the y axis 2) Check	
		that the loo/conil method ap	
		plies to both area 4) Check that	
		the re built object is the same as	
		the re-built object is the same as $\frac{1}{2}$	
		111 Out[13].	
09	Lest the snape of the data in	less that the [ao/conj] method	pass
	AOS.	keeps the data snape of the input	
		object. The input AO data must	
		be an array with row data and/or	
		column data.	
		1) Check that the shape of the	pass
		data doesn't change.	
10	Check that the [ao/conj]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/conj]	Call the method with a list of	pass
1 1 1	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	
10	Check that the errors are	Call the method with a list of	pass
12	cleared for this method.	output variables and with a sin-	
		gle output variable. Additionaly	
		check that the rebuild method	
		works on the output.	
		1) Check that the output has no	pass
		error fields	·

Table 16: Unit tests for ao/conj.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/conv			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists	pass
		have the correct parameters.	
02	Tests that the conv method	Test that the conv method works	pass
	works with a vector of AOs as	for a vector of AOs as input.	
	input.	1) Check that the output is exact	pass
		one Ao with cdata. 2) Check that	
		each output AO contains the cor-	
		rect data.	
03	Tests that the conv method	Test that the conv method works	pass
	works with a matrix of AOs as	for a matrix of AOs as input.	
	input.	1) Check that the output is exact	pass
		one Ao with cdata. 2) Check that	
		each output AO contains the cor-	
		rect data.	
04	Tests that the conv method	Test that the conv method works	pass
	works with a list of AOs as	for a list of AOs as input.	
	input.	1) Check that the output is exact	pass
		one Ao with cdata. 2) Check that	
		each output AO contains the cor-	
		rect data.	
05	Tests that the conv method	Test that the conv method works	pass
	works with a mix of different	with an input of matrices and	
	snaped AOs as input.	vectors and single AOs.	
		1) Check that the output is exact	pass
		one Ao with cdata. 2) Check that	
		each output AO contains the cor-	
	Tests that the conversethed	Tect data.	
06	rests that the convinction	ing the correspondence has pro-	pass
	property applies instory.	ing the convinethod can be pro-	
		1) Check that the last entry in	nare
		the history of 'out' corresponds	pass
		to 'conv' 2) Check that the re-	
		built object is the same object as	
		the input	
	The conv method can not	The conv method throws an or	naee
07	modify the input AO	ror if it is used as a modifier	pass
	moury the input rio.	1) Nothing to test	nass
<u> </u>	Check that the [ao/conv]	Call the method with a list of	1 pass
11	method pass back the output	output variables and with a sin-	pass
	objects to a list of output	gle output variable Additionaly	
	variables or to a single variable	check that the rebuild method	
		works on the output	
		1) Check that the output con-	nass
		tains the same plotinfo plist	Pass
	I I	1 one serie provinito prov	1



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/conv

Table 17: Unit tests for a o/conv.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/convert			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the convert method works with a vector of AOs as input.	Test that the convert method works for a vector of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atvec' 2) Check that each output AO contains the correct data.	pass
03	Tests that the convert method works with a matrix of AOs as input.	Test that the convert method works for a matrix of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atmat' 2) Check that each output AO contains the correct data.	pass
04	Tests that the convert method works with a list of AOs as input.	Tests that the convert method works with a list of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atmat' 2) Check that each output AO contains the correct data.	pass
05	Tests that the convert method works with a mix of different shaped AOs as input.	Tests that the convert method works with a mix of different shaped AOs as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atmat' 2) Check that each output AO contains the correct data.	pass
06	Tests that the convert method properly applies history.	Test that the result of applying the convert method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'convert'. 2) Check that the re- built object is the same object as the input.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/convert			
07	Tests that the convert method	Test that the convert method can	pass
07	can modify the input AO.	modify the input AO by call-	
		ing with no output and that the	
		method doesn't change the input	
		of the function notation (with a	
		equal sign).	
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	1
		is not changed 3) Check that	
		the modified input is the convert	
		value of the copy 4) Check that	
		out and amodi are the same	
		Test that the convert method can	pass
08	Control the method with a plist.	modify the input object depend-	Pass
		ing to the plist.	
		1) Check that the convert	pass
		method appressible different	P
		actions 4) Check that the re-	
		built objects are the same as	
		$\operatorname{out}[16]$	
		Test that the convert method	pass
09	Test the shape of the output.	keeps the data shape of the in-	Pass
		put object. The input AO must	
		be an AO with row data and an	
		AQ with column data	
		1) Check that the shape of the	pass
		data doesn't change.	Pass
	Check that the convert method	Call the method with a list of	pass
10	pass back the output objects to	output variables and with a sin-	Pass
	a list of output variables or to a	gle output variable Additionaly	
	single variable	check that the rebuild method	
	single variable.	works on the output	
		1) Check that the output con-	nass
		tains the right number of ob-	Pass
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'	
	Check that the [ao/convert]	Call the method with a list of	nase
11	method pass back the output	output variables and with a sin-	Pass
	objects to a list of output	gle output variable Additionaly	
	variables or to a single variable	check that the rebuild method	
	variables of to a single variable.	works on the output	
		1) Check that the output con	nase
		tains the same plotinfo plict	pass
		vains inc same provinto pust	

Table 18: Unit tests for ao/convert.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/copy			
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	

Table 19: Unit tests for a o/copy.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/cos			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the [ao/cos] method	Test that the [ao/cos] method	pass
02	works with a vector of objects	works for a vector of objects as	
	as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'vec' 2) Check that each output	
		object contains the correct data.	
03	Tests that the [ao/cos] method	Test that the [ao/cos] method	pass
	works with a matrix of objects	works for a matrix of objects as	
	as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
04	Tests that the [ao/cos] method	Test that the [ao/cos] method	pass
	works with a list of objects as	works for a list of objects as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		(mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/cos] method	Tests that the [ao/cos] method	pass
	works with a mix of different	works with a mix of different ar-	
	arrays of objects as input.	rays of objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		abject contains the correct data	
	Tests that the les (see) method	The trans the correct data.	magg
06	properly applies history	the loc /cocl method can be pro-	pass
	property applies instory.	accord back	
		1) Check that the last entry in	negg
		the history of 'out' corresponde	pass
		the instory of out corresponds	
		ro built object is the same object	
		as the input	
	Tests that the [ac/cos] method	Test that the [ao/cos] method	nase
07	can modify the input AO	can modify the input object by	pass
	can mouny the input AO.	calling with no output and that	
		the method doesn't change the	
		input of the function notation	
		(with a equal sign)	
		(with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/cos			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	-
		is not changed 3) Check that	
		the modified input is the [ao/cos]	
		value of the copy 4) Check that	
		out and amodi are the same	
	Test that the [ao/cos] method	Test that the [ao/cos] method	pass
08	uses the plist to get the axis.	uses the plist to get the axis.	-
		1) Check that the $[ao/cos]$	pass
		method applies to the x-axis 2)	-
		Check that the [ao/cos] method	
		applies to the y-axis 3) Check	
		that the [ao/cos] method applies	
		to both axes 4) Check that the	
		re-built object is the same as in	
		'out[13]'.	
00	Test the shape of the data in	Test that the [ao/cos] method	pass
09	AOs.	keeps the data shape of the in-	-
		put object. The input AO data	
		must be an array with row data	
		and/or column data.	
		1) Check that the shape of the	pass
		data doesn't change.	-
10	Check that the [ao/cos] method	Call the method with a list of	pass
10	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/cos] method	Call the method with a list of	pass
11	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 20: Unit tests for ao/cos.


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/cov			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases, 2) Check that all plists	
		have the correct parameters.	
	Tests that the cov method works	Test that the cov method works	nass
02	with a vector of AOs as input	for a vector of AOs as input	pass
	with a vector of field as input.	1) Check that the output is exact	nass
		one Ao with cdata 2) Check that	pass
		each output AO contains the cor-	
		rect data	
	Tosts that the cov method works	Test that the cov method works	narr
03	with a matrix of AOs as input	for a matrix of $\Delta \Omega_{\rm S}$ as input	pass
	with a matrix of AOS as input.	1) Check that the output is event	pagg
		one Ac with adate 2) Check that	pass
		one Ao with cuata. 2) Check that	
		each output AO contains the col-	
	Trata that the same mathed membra	Tect data.	
04	Tests that the cov method works	Tests that the cov method works	pass
	with a list of AOs as input.	1) Charlet hat the sector is seen at	
		1) Check that the output is exact	pass
		one Ao with cdata. 2) Check that	
		each output AO contains the cor-	
		rect data.	
05	Tests that the cov method	Tests that the cov method works	pass
	works with a mix of different	with a mix of different snaped	
	shaped AOs as input.	AOs as input.	
		1) Check that the output is exact	pass
		one Ao with cdata. 2) Check that	
		each output AO contains the cor-	
		rect data.	
06	Tests that the cov method	Test that the result of applying	pass
	properly applies history.	the cov method can be processed	
		back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'cov'. 2) Check that the re-	
		built object is the same object as	
		the input.	
07	The cov method can not modify	The cov method can not modify	pass
	the input AO.	the input AO.	
		1) Nothing to test.	pass
11	Check that the [ao/cov] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/cov

Table 21: Unit tests for ao/cov.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/cpsd			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.0	Tests that the cpsd method	Test that the cpsd method works	pass
02	works with a vector of AOs as	for a vector of AOs as input.	
	input. (only with two objects in	1) Check that the number of el-	pass
	the vector)	ements in 'out' is the square of	
		the number in the input.	
0.2	Tests that the cpsd method	Test that the cpsd method	pass
05	doesn't work with a matrix of	doesn't work for a matrix of AOs	
	AOs as input.	as input.	
		1) Nothing to check.	pass
04	Tests that the cpsd method	Test that the cpsd method works	pass
04	works with a list of AOs as	for a list of AOs as input.	
	input.	1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
05	Tests that the cpsd method	Test that the cpsd method	pass
00	doesn't work with a mix of	doesn't work with an input of	
	different shaped AOs as input.	matrices and vectors and single	
		AOs.	
		1) Nothing to check	pass
06	Tests that the cpsd method	Test that the result of apply-	pass
00	properly applies history.	ing the cpsd method can be pro-	
		cessed back to an m-file.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'cpsd'. 2) Check that the re-	
		built object is the same as 'out'.	
07	Tests that the cpsd method can	Test that the cpsd method can	pass
	not modify the input AO.	not modify the input AO. The	
		method must throw an error for	
		the modifier call.	
		1) Nothing to check.	pass
08	Test the shape of the output.	1 lest that the cpsd method keeps	pass
		the data shape of the input ob-	
		ject. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shpe of the	pass
		output data doesn't change.	
09	Check that the cpsd method	This test is not longer necessary	pass
	pass back the output objects to	because the cpsd method pass	
	a list of output variables or to a	back always only one object.	
	single variable.		



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/cpsd			
		1) Nothing to check.	pass
10	Tests that the cpsd method	Test that the applying cpsd	pass
10	agrees with MATLAB's cpsd	works on two AOs.	
	when configured to use the same	1) Check that output agrees with	pass
	parameters.	the output of MATLAB's cpsd.	
11	Check that the [ao/cpsd]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	
17	Tests handling of units: 1) white	1) Prepare the test tsdata: white	pass
11	noise produced from normal	noise from normal distribution +	
	pdf, with a given mean value	offset 2) Assign a random unit	
	and sigma (distribution's 1st	3) Prepare the test tsdata: white	
	and 2nd orders) 2) white noise	noise from normal distribution +	
	produced from normal pdf, with	offset 4) Assign a random unit 5)	
	a given mean value and sigma	CPSD of the white noise	
	(distribution's 1st and 2nd	1) Check that (calculated	pass
	orders) 3) CPSD of the white	CPSD yunits) equals input_1	
	noise series 4) compares the	$units*input_2 units/Hz$	
18	Tests datablingport amidsout)produite	1) Prepare the test tsdata: white	pass
10	noise produced from normal	noise from normal distribution +	
	pdf, with a given mean value	offset 2) Assign a random unit	
	and sigma (distribution's 1st	3) Prepare the test tsdata: white	
	and 2nd orders) 2) white noise	noise from normal distribution +	
	produced from normal pdf, with	offset 4) Assign a random unit 5)	
	a given mean value and sigma	CPSD of the white noise 6) PSD	
	(distribution's 1st and 2nd	of the white noise	
	orders) 3) CPSD of the white	1) Check that $CPSD(x,y)$ equals	pass
	noise series Comparison with	conj(CPSD(y,x)) 2) Check that	
	PSD: 4) compares the	CPSD(x,x) equals $PSD(x) = 3$	
	off-diagonal terms to check they	Check that CPSD(y,y) equals	
	are complex-conjugated 5)	PSD(y)	
24	Testpatesttdiffdiagohyasizedmsata	Test that applying cpsd works on	pass
	seitshare treateble properdy al noise	two AOs.	
		1) Check that cpsd used the	pass
		length of the shortest ao.	
25	Tests handling of units: 1) white	1) Prepare the test tsdata: white	pass
	noise produced from normal	noise from normal distribution +	
	pdf, with a given mean value	offset 2) Assign a random unit	
	and sigma (distribution's 1st	3) Prepare the test tsdata: white	
	and 2nd orders) 2) white noise	noise from normal distribution +	
	produced from normal pdf, with	offset 4) Assign a random unit 5)	
	a given mean value and sigma	CPSD of the white noise	
	(distribution's 1st and 2nd	1) Check that (calculated	pass
	orders) 3) CPSD of the white	CPSD yunits) equals input_1	
	noise series 4) compares the	units*input_2 units/Hz	

units of the input and output



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/cpsd			
51	Tests the possibility to set the	1) Prepare the test tsdata: white	pass
01	number of averages rather than	noise from normal distribution $+$	
	setting the Nfft: 1) white noise	offset 2) cpsd of the noise, with-	
	produced from normal pdf,	out detrending, random window,	
	with: a given mean value and	set number of averages	
	sigma (distribution's 1st and	1) Check that calculated navs are	pass
	2nd order) 2) cpsd of the noise,	identical to those requested	
50	Teshouheletossidility tonstanthe	1) white noise produced from	pass
52	window, stet verngløsrraf lævertages	uniform pdf, with: a given mean	
	3)tchagkthedveffect)vohitenboisof	value and sigma (distribution's	
	prortages d from uniform pdf,	1st and 2nd order) 2) cpsd of the	
	with: a given mean value and	noise, without detrending, ran-	
	sigma (distribution's 1st and	dom window, random navs 3) get	
	2nd order) 2) cpsd of the noise,	the number of averages 4) get the	
	without detrending, random	nfft used 5) run cpsd again, with	
	window, random navs 3) get the	the nfft used	
	number of averages 4) get the	1) Check that calculated objects	pass
	nfft used 5) run cpsd again,	C1 and C2 are identical	
53	Weststheenflissiseility) toosepahe	1) white noise produced from	pass
00	then be culated raises tather than	normal pdf, with: a given mean	
	setting the Nfft: 1) white noise	value and sigma (distribution's	
	produced from normal pdf,	1st and 2nd order) 2) cpsd of the	
	with: a given mean value and	noise, without detrending, ran-	
	sigma (distribution's 1st and	dom window, random navs 3) get	
	2nd order) 2) cpsd of the noise,	the number of averages 4) get	
	without detrending, random	the nfft used 5) run cpsd again,	
	window, random navs 3) get the	with the nfft used 6) run cpsd	
	number of averages 4) get the	again, with conflicting parame-	
	nfft used 5) run cpsd again,	ters, and verify it uses nfft rather	
	with the nfft used 6) compare	than navs	
	navs, nfft, psds	1) Check that calculated objects	pass
		C1 and C2 are identical 2) Check	
		that C3 used different values	

Table 22: Unit tests for ao/cpsd.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/created			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the created method works with a vector of AOs as input.	Test that the created method works for a vector of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atvec' 2) Check that each output contains the correct data.	pass
03	Tests that the created method works with a matrix of AOs as input.	Test that the created method works for a matrix of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atmat' 2) Check that each out- put contains the correct data.	pass
04	Tests that the created method works with a list of AOs as	Test that the created method works for a list of AOs as input.	pass
	mput.	ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the created method works with a mix of different shaped AOs as input.	Test that the created method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the created method properly applies history	This method doesn't change the input object, thus no history is added to the object.	pass
07	Tests that the created method can be used with the modify command.	Tests that the created method can be used with the modify command.	pass
		1) Check the single object 2) Check the matrix object	pass
08	Tests that the created method retruns always a well defined	Test that the created method with an empty AO	pass
	time object even for an empty input object.	1) Check that the output is a time object with a ell defined time.	pass

Table 23: Unit tests for a o/created.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/creator			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists	pass
		nave the correct parameters.	
02	works with a vector of AOs as	vorks for a vector of AOs as in-	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atvec' 2) Check that each output contains the correct data.	pass
03	Tests that the creator method works with a matrix of AOs as input.	Test that the creator method works for a matrix of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atmat' 2) Check that each out- put contains the correct data.	pass
04	Tests that the creator method works with a list of AOs as	The creator method doesn't work for a list of AOs as input.	pass
05	Tests that the creator method works with a mix of different shaped AOs as input.	The creator method doesn't work with different shaped input ob- jects.	pass
		1) Nothing to test	pass
06	Tests that the creator method properly applies history	This method doesn't change the input object, thus no history is added to the object.	pass
		1) Nothing to check.	pass
07	Tests that the creator method can be used with the modify command.	Tests that the creator method can be used with the modify command.	pass
		1) Check the single object 2) Check the matrix object	pass
08	Tests that the creator method retruns all creator(s)/modifier(s) which are in the history.	Test that the creator method uses the option 'all' direct or in a plist. The test file must have the modifier 'first', 'second' and 'third'	pass
		1) Check that out1 contains only one creator 2) Check that out2 contain more creator/modifier	pass
09	Tests the negative case for the option 'all'.	Test that the creator method throws an error if the option 'all' is used in connection with a ma- trix/vector of AOs.	pass
		1) Nothing to test.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/creator

Table 24: Unit tests for a o/creator.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/ctranspose			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the [ao/ctranspose]	Test that the [ao/ctranspose]	pass
	method works with a vector of	method works for a vector of ob-	
	objects as input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'vec' 2) Check that each output	
		object contains the correct data.	
03	Tests that the [ao/ctranspose]	Test that the [ao/ctranspose]	pass
	method works with a matrix of	method works for a matrix of ob-	
	objects as input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
04	Tests that the [ao/ctranspose]	Test that the [ao/ctranspose]	pass
04	method works with a list of	method works for a list of objects	
	objects as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/ctranspose]	Tests that the [ao/ctranspose]	pass
	method works with a mix of	method works with a mix of dif-	
	different arrays of objects as	ferent arrays of objects as input.	
	input.	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
06	Tests that the [ao/ctranspose]	Test that the result of applying	pass
	method properly applies history.	the [ao/ctranspose] method can	
		be processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[ao/ctranspose]'. 2) Check	
		that the re-built object is the	
		same object as the input.	
07	Tests that the ctranspose	Test that the ctranspose method	pass
	method can modify the input	can modify the input AO by call-	
	AO.	ing with no output and that the	
		method doesn't change the input	
		of the function notation (with a	
		equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/ctranspose			
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		is ctranspose(at1).	
00	Control the method with a plict	Test that the abs method keeps	pass
09	Control the method with a plist.	the data shape of the input ob-	
		ject. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
10	Check that the [ao/ctranspose]	Call the method with a list of	pass
10	method pass back the output	output variables and with a sin-	-
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
	Ŭ	works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	1
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/ctranspose]	Call the method with a list of	pass
11	method pass back the output	output variables and with a sin-	1
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	
10	Check that the errors are	Call the method with a list of	pass
12	cleared for this method.	output variables and with a sin-	1
		gle output variable. Additionaly	
		check that the rebuild method	
		works on the output.	
		1) Check that the output has no	pass
		error fields	1 1 1 1 1 1 1

Table 25: Unit tests for ao/ctranspose.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/delay			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually	pass
		1) Check that got Info call ro	nase
		turned on minfo object in all	pass
		cases 2) Check that all plists	
		have the correct parameters	
	Tests that the [ac/delay]	Test that the lee (delay) method	pagg
02	method works with a vestor of	werka for a water of objects as	pass
	abiests as input	works for a vector of objects as	
	objects as input.	1) Check that the number of al	nagg
		i) Check that the humber of el-	pass
		'men' 2) Check that each output	
		object contains the correct data	
	Tests that the [ac/delay]	Test that the lee (delay) method	pagg
03	method works with a metric of	werks for a matrix of objects as	pass
	abiests as input	works for a matrix of objects as	
	objects as input.	1) Check that the number of al	pagg
		1) Check that the humber of el-	pass
		'mat' 2) Chask that each sutput	
		abiest contains the correct date	
		The table to the first the correct data.	
04	Tests that the [ao/delay]	Test that the [ao/delay] method	pass
	method works with a list of	works for a list of objects as in-	
	objects as input.	1) Check that the number of al	pagg
		1) Check that the humber of el-	pass
		'mat' 2) Chask that each sutput	
		abiest contains the correct date	
	Tests that the [as /delaw]	Tests that the [as /delaw] method	magg
05	method works with a min of	rests that the [ao/delay] method	pass
	different errors of chiests of	works with a mix of different ar-	
	different arrays of objects as	1) Check that the number of al	magg
	mput.	1) Check that the humber of el-	pass
		ements in out is the same as in	
		abiest contains the correct date	
	Tests that the [as /delaw]	Test that the result of apply	magg
06	method property applies history	ing the less (delay) method can be	pass
	method property applies history.	mig the [ao/delay] method can be	
		1) Check that the last entry in	magg
		1) Check that the last entry in the history of 'out' company and	pass
		the history of out corresponds	
		to [ao/delay]. 2) Check that the	
		as the input	
	Tests that the [/-l-l]	The as the input.	
07	rests that the [ao/delay]	1 lest that the [ao/delay] method	pass
	method can modify the input	can modify the input object by	
	AU.	canning with no output and that	
		the method doesn't change the	
		input of the function notation	
		(with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/delay			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the [ao/delay]	
		value of the copy 4) Check that	
		out and amodi are the same	
00	Test the shape of the data in	Test that the [ao/delay] method	pass
09	AOs.	keeps the data shape of the input	
		object. The input AO data must	
		be an array with row data and/or	
		column data.	
		1) Check that the shape of the	pass
		data doesn't change.	-
10	Check that the [ao/delay]	Call the method with a list of	pass
10	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	-
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/delay]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	
10	Check that the errors are	Call the method with a list of	pass
12	cleared for this method.	output variables and with a sin-	
		gle output variable. Additionaly	
		check that the rebuild method	
		works on the output.	
		1) Check that the output has no	pass
		error fields	

Table 26: Unit tests for a o/delay.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/demux			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the demux method	Tests that the demux method	pass
02	works with a mix of different	works with a mix of different	
	shaped AOs as input.	shaped AOs as input.	
		1) Check the output objects.	pass
02	Negative test. Check that the	Check that the demux method	pass
05	demux method throwns an error	throwns an error for too few out-	
	for too few output variables.	put variables.	
		1) Nothing to check.	pass
0.4	Negative test. Check that the	Check that the demux method	pass
04	demux method throwns an error	throwns an error for too few out-	
	for too many output variables.	put variables.	
		1) Nothing to check.	pass

Table 27: Unit tests for ao/demux.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/det			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the det method works with a vector of AOs as input.	Test that the det method works for a vector of AOs as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atvec' 2) Check that each output AO contains the correct data.	pass
03	Tests that the det method works with a matrix of AOs as input.	Test that the det method works for a matrix of AOs as input. 1) Check that the number of el-	pass
		ements in 'out' is the same as in 'atmat' 2) Check that each output AO contains the correct data.	I man
04	Tests that the det method works with a list of AOs as input.	Test that the det method works for a list of AOs as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
05	Tests that the det method works with a mix of different shaped AOs as input.	Test that the det method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the det method properly applies history.	Test that the result of applying the det method can be processed back to an m-file.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'det'. 2) Check that the re- built object is the same object as 'out'.	pass
07	Tests that the det method can modify the input AO.	Test that the det method can modify the input AO by call- ing with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass
		1) Check that 'at4' and 'ain' are now different. 2) Check that 'ain' is det(at4).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/det			
08	Control the method with a plist	Test that the det method can	pass
	Control the method with a plist.	modify the single axis controlled	
		by the plist and the resuld can be	
		processed back to an m-file.	
		1) Check that the det method	pass
		applies to the x-axis 2) Check	
		that the det method applies to	
		the y-axis 3) Check that the det	
		method applies to both axes 4)	
		Check that the re-built objects	
		are the same object as ' $out[13]$ '.	
00	Control the method with a plist	Test that the det method keeps	pass
05	Control the method with a plist.	the data shape of the input ob-	
		ject. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
10	Check that the det method pass	Call the method with a list of	pass
	back the output objects to a list	output variables and with a sin-	
	of output variables or to a single	gle output variable. Additionaly	
	variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/det] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	
12	Check that the errors are	Call the method with a list of	pass
	cleared for this method.	output variables and with a sin-	
		gle output variable. Additionaly	
		check that the rebuild method	
		works on the output.	
		1) Check that the output has no	pass
		error fields	

Table 28: Unit tests for ao/det.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/detrend			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the [ao/detrend]	Test that the [ao/detrend]	pass
02	method works with a vector of	method works for a vector of	
	objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'vec' 2) Check that each output	
		object contains the correct data.	
03	Tests that the [ao/detrend]	Test that the [ao/detrend]	pass
00	method works with a matrix of	method works for a matrix of	
	objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
04	Tests that the [ao/detrend]	Test that the [ao/detrend]	pass
04	method works with a list of	method works for a list of	
	objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/detrend]	Tests that the [ao/detrend]	pass
00	method works with a mix of	method works with a mix of	
	different arrays of objects as	different arrays of objects as	
	input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
06	Tests that the [ao/detrend]	Test that the result of applying	pass
	method properly applies history.	the [ao/detrend] method can be	
		processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[ao/detrend]'. 2) Check that	
		the re-built object is the same	
		object as the input.	
07	Tests that the [ao/detrend]	Test that the [ao/detrend]	pass
01	method can modify the input	method can modify the input	
	AO.	object by calling with no output	
		and that the method doesn't	
		change the input of the function	
		notation (with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/detrend			
		1) Check that 'out' and 'aeq'	pass
		are now different. 2) Check that	
		'aeq' is not changed 3) Check	
		that the modified input is the	
		[ao/detrend] value of the copy 4)	
		Check that out and amodi are	
		the same	
00	Test the shape of the data in	Test that the [ao/detrend]	pass
09	AOs.	method keeps the data shape of	
		the input object. The input AO	
		data must be an array with row	
		data and/or column data.	
		1) Check that the shape of the	pass
		data doesn't change.	-
10	Check that the [ao/detrend]	Call the method with a list of	pass
10	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	1
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/detrend]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	
10	Check that the errors are	Call the method with a list of	pass
12	cleared for this method.	output variables and with a sin-	
		gle output variable. Additionaly	
		check that the rebuild method	
		works on the output.	
		1) Check that the output has no	pass
		error fields	

Table 29: Unit tests for ao/detrend.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/dft			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minto object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the [ao/dft] method	lest that the [ao/dft] method	pass
	works with a vector of objects	works for a vector of objects as	
	as input.	input.	
		1) Check that the number of el-	pass
		ements in out is the same as in	
		vec 2) Check that each output	
	Tests that the [as /dft] method	The that the log dft method	
03	rests that the [ao/dit] method	rest that the [ao/dit] method	pass
	works with a matrix of objects	works for a matrix of objects as	
	as input.	1) Check that the number of el	page
		oments in 'out' is the same as in	pass
		'mat' 2) Check that each output	
		object contains the correct data	
	Tests that the [ao/dft] method	Test that the [ao/dft] method	nass
04	works with a list of objects as	works for a list of objects as in-	pass
	input	but	
	mpuo.	1) Check that the number of el-	pass
		ements in 'out' is the same as in	pass
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/dft] method	Tests that the [ao/dft] method	pass
05	works with a mix of different	works with a mix of different ar-	
	arrays of objects as input.	rays of objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
06	Tests that the [ao/dft] method	Test that the result of applying	pass
	properly applies history.	the [ao/dft] method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		$\int to '[ao/dft]'$. 2) Check that the	
		re-built object is the same object	
		as the input.	
07	Tests that the [ao/dft] method	Test that the [ao/dft] method	pass
	can modify the input AO.	can modify the input object by	
		calling with no output and that	
		the method doesn't change the	
		input of the function notation	
		(with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/dft			
,		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	-
		is not changed 3) Check that	
		the modified input is the [ao/dft]	
		value of the copy 4) Check that	
		out and amodi are the same	
	Test the shape of the data in	Test that the [ao/dft] method	pass
09	AOs	keeps the data shape of the in-	P
		put object. The input AO data	
		must be an array with row data	
		and/or column data	
		1) Check that the shape of the	nase
		data doesn't change	pass
	Check that the [ao/dft] method	Call the method with a list of	nase
10	pass back the output objects to	output variables and with a sin-	pass
	a list of output variables or to a	de output variable. Additionaly	
	a list of output variables of to a	sheels that the populational	
	single variable.	ments on the output	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the rebuild	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/dft] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	
10	Check that the errors are	Call the method with a list of	pass
14	cleared for this method.	output variables and with a sin-	
		gle output variable. Additionaly	
		check that the rebuild method	
		works on the output.	
		1) Check that the output has no	pass
		error fields	

Table 30: Unit tests for ao/dft.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/diag			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the diag method works with a vector of AOs as	Test that the diag method works for a vector of AOs as input.	pass
	input.	1) Check that the number of el- ements in 'out' is the same as in 'atvec' 2) Check that each output AO contains the correct data.	pass
03	Tests that the diag method works with a matrix of AOs as	Test that the diag method works for a matrix of AOs as input.	pass
	input.	ements in 'out' is the same as in 'atmat' 2) Check that each output AO contains the correct data.	pass
04	Tests that the diag method works with a list of AOs as input.	Test that the diag method worksfor a list of AOs as input.1) Check that the number of el-	pass pass
		ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	
05	Tests that the diag method works with a mix of different shaped AOs as input.	Test that the diag method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the diag method properly applies history.	Test that the result of apply- ing the diag method can be pro- cessed back to an m-file.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'diag'. 2) Check that the re- built object is the same object as 'out'.	pass
07	Tests that the diag method can modify the input AO.	Test that the diag method can modify the input AO by call- ing with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass
		1) Check that 'at4' and 'ain' are now different. 2) Check that 'ain' is diag(at4).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/diag			
08	Control the method with a plist.	Test that the diag method can modify the single axis controlled by the plist and the resuld can be processed back to an m-file.	pass
		1) Check that the svd method applies with different options. 2) Check that the re-built object is the same object as 'out'.	pass
09	Control the method with a plist.	Test that the diag method keeps the data shape of the input ob- ject. The input AO must be an AO with row data and an AO with column data.	pass
		1) Check that the shpe of the data doesn't change.	pass
10	Check that the diag method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionally check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
11	Check that the [ao/diag] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the same plotinfo plist	pass
12	Check that the errors are cleared for this method.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		error fields	pass

Table 31: Unit tests for a o/diag.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/diff			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.9	Tests that the [ao/diff] method	Test that the [ao/diff] method	pass
02	works with a vector of objects	works for a vector of objects as	
	as input.	input.	
	-	1) Check that the number of el-	pass
		ements in 'out' is the same as in	1
		'vec' 2) Check that each output	
		object contains the correct data.	
	Tests that the [ao/diff] method	Test that the [ao/diff] method	pass
03	works with a matrix of objects	works for a matrix of objects as	1
	as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	pass
		'mat' 2) Check that each output	
		object contains the correct data	
	Tosts that the [ao/diff] method	Tost that the [ao/diff] method	nage
04	works with a list of objects as	works for a list of objects as in	pass
	input	works for a fist of objects as fif-	
	mput.	1) Check that the number of el	nase
		amonts in 'out' is the same as in	pass
		'mat' 2) Check that each output	
		abject contains the correct data	
	Tests that the [as/diff] method	Tests that the leg/diff] method	naga
05	works with a mix of different	works with a mix of different an	pass
	works with a mix of different	works with a mix of different ar-	
	arrays of objects as input.	1) Check that the merch of all	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		mat ²) Check that each output	
		object contains the correct data.	
06	lests that the [ao/diff] method	Test that the result of applying	pass
	properly applies history.	the [ao/diff] method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to $'[ao/diff]'$. 2) Check that the	
		re-built object is the same object	
		as the input.	
07	Tests that the [ao/diff] method	Test that the [ao/diff] method	pass
	can modify the input AO.	can modify the input object by	
		calling with no output and that	
		the method doesn't change the	
		input of the function notation	
		(with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/diff			
		1) Check that 'out' and 'aeq' are now different. 2) Check that 'aeq' is not changed 3) Check that the modified input is the [ao/diff] value of the copy 4) Check that out and amodi are the same	pass
09	Test the shape of the data in AOs.	 Test that the [ao/diff] method keeps the data shape of the input object. The input AO data must be an array with row data and/or column data. 1) Check that the shape of the data doesn't change. 	pass
10	Check that the [ao/diff] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
11	Check that the [ao/diff] method pass back the output objects to a list of output variables or to a single variable.	 Call the method with a list of output variables and with a single output variable. Additionally check that the rebuild method works on the output. 1) Check that the output con- 	pass
12	Check that the errors are cleared for this method.	 tains the same plotinfo plist Call the method with a list of output variables and with a single output variable. Additionaly check that the rebuild method works on the output. 1) Check that the output has possible. 	pass
13	Control the method with a plist.	 1) Check that the output has no error fields Test the computation of derivative using a 2nd order 1) Check that the diff method uses the 2nd order derivative and and and and and and and and and and	pass pass
14	Control the method with a plicit	Uses the 2nd order derivative. 2) Check that the re-built object is the same object as 'out'. Test the computation of deriva-	pass
	Control the method with a plist.	tive using a 2nd order with a parabolic fit	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/diff			
		1) Check that the diff method	pass
		uses the 2nd order derivative	
		with a parabolic fit 2) Check that	
		the re-built object is the same	
		object as 'out'.	
15	Control the method with a plist	Test the 5 point derivative.	pass
10	Control the method with a plist.	1) Check that the diff method	pass
		uses the 5 point derivative. 2)	
		Check that the re-built object is	
		the same object as 'out'.	

Table 32: Unit tests for ao/diff.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/display			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the display method	Test that the display method	pass
02	works with a vector of AOs as	works for a vector of AOs as in-	
	input.	put.	
		1) Check that the output contain	pass
		at least each object name	
03	Tests that the display method	Test that the display method	pass
05	works with a matrix of AOs as	works for a matrix of AOs as in-	
	input.	put.	
		1) Check that the output contain	pass
		at least each object name	
04	Tests that the display method	Test that the display method	pass
04	works with a list of AOs as	works for a list of AOs as input.	
	input.	1) Check that the output contain	pass
		at least each object name	
05	Tests that the display method	Test that the display method	pass
00	works with a mix of different	works with an input of matrices	
	shaped AOs as input.	and vectors and single AOs.	
		1) Check that the output contain	pass
		at least each object name	
06	Tests that the display method	The method display doesn't	pass
00	properly applies history.	change the data, thus it is not	
		possible to check the history.	
		Nothing to do.	
			pass

Table 33: Unit tests for ao/display.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/dopplercorr			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the dopplercorr method works with a vector of AOs as input.	Test that the dopplercorr method works for a vector of AOs as input.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
03	Tests that the dopplercorr method works with a matrix of AOs as input.	Test that the dopplercorr method works for a matrix of AOs as input.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
04	Tests that the dopplercorr method works with a list of AOs as input.	Test that the dopplercorr method works for a list of AOs as input.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
05	Tests that the dopplercorr method works with a mix of different shaped AOs as input.	Test that the dopplercorr method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the dopplercorr method properly applies history.	Test that the result of applying the dopplercorr method can be processed back to an m-file.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'dopplercorr'. 2) Check that the re-built object is the same object as 'out'.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/dopplercorr			
07	Check that the dopplercorr method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
11	Check that the [ao/dopplercorr] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the same plotinfo plist	pass

Table 34: Unit tests for a o/dopplercorr.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/downsample			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the downsample method works with a vector of AOs as input.	Test that the downsample method works for a vector of AOs as input.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
03	Tests that the downsample method works with a matrix of AOs as input.	Test that the downsample method works for a matrix of AOs as input.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
04	Tests that the downsample method works with a list of AOs as input.	Test that the downsample method works for a list of AOs as input.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
05	Tests that the downsample method works with a mix of different shaped AOs as input.	Test that the downsample method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the downsample method properly applies history.	Test that the result of applying the downsample method can be processed back to an m-file.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'downsample'. 2) Check that the re-built object is the same object as 'out'.	pass
07	Tests that the downsample method can modify the input AO.	Test that the downsample method can modify the input AO by calling with no output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/downsample			
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		is downsample(at1).	
08	Tests that the downsample	Test that the downsample	pass
08	method keeps the data shape of	method keeps the data shape of	
	the input object.	the input object. The input AO	
		must be an AO with row data	
		and an AO with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	-
00	Check that the downsample	Call the method with a list of	pass
09	method pass back the output	output variables and with a sin-	-
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	-
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
10		Test the downsample method	pass
10	Control the method with a plist.	with a factor and an offset.	-
		1) Check that the downsample	pass
		method with an offset and a fac-	
		tor 2) Check that the re-built ob-	
		ject is the same object as 'out'.	
11	Check that the [ao/downsample]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	
10	Check that the errors are	Call the method with a list of	pass
	cleared for this method.	output variables and with a sin-	
		gle output variable. Additionaly	
		check that the rebuild method	
		works on the output.	
		1) Check that the output has no	pass
		error fields	

Table 35: Unit tests for ao/downsample.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/dropduplicates			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters	pass
02	Tests that the dropduplicates method works with a vector of AOs as input.	Test that the dropduplicates method works for a vector of AOs as input.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
03	Tests that the dropduplicates method works with a matrix of AOs as input.	Test that the dropduplicates method works for a matrix of AOs as input.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
04	Tests that the dropduplicates method works with a list of AOs as input.	Test that the dropduplicates method works for a list of AOs as input.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
05	Tests that the dropduplicates method works with a mix of different shaped AOs as input.	Test that the dropduplicates method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the dropduplicates method properly applies history.	Test that the result of applying the dropduplicates method can be processed back to an m-file.	pass
		1) Check that the last entry in the history of 'out' corre- sponds to 'dropduplicates'. 2) Check that the re-built object is the same object as 'out'.	pass
07	Tests that the dropduplicates method can modify the input AO.	Test that the dropduplicates method can modify the input AO by calling with no output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/dropduplicates			
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		is dropduplicates(at1).	
08	Tests that the dropduplicates	Test that the dropduplicates	pass
	method keeps the data shape of	method keeps the data shape of	
	the input object.	the input object. The input AO	
		must be an AO with row data	
		and an AO with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
09	Check that the dropduplicates	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
10	Control the method with a plist	Test the dropduplicates method	pass
10	Control the method with a plist.	with different tolerances.	
		1) Check that the different toler-	pass
		ances 2) Check that the re-built	
		object is the same object as 'out'.	
11	Check that the	Call the method with a list of	pass
11	[ao/dropduplicates] method	output variables and with a sin-	
	pass back the output objects to	gle output variable. Additionaly	
	a list of output variables or to a	check that the rebuild method	
	single variable.	works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 36: Unit tests for ao/dropduplicates.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/dsmean			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the dismean method works with a vector of AOs as input.	Test that the dsmean method works for a vector of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
03	Tests that the dismean method works with a matrix of AOs as input.	Test that the dsmean method works for a matrix of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
04	Tests that the dismean method works with a list of AOs as input.	Test that the dsmean method works for a list of AOs as input. 1) Check that the number of ele- ments in 'out' is the square of the	pass pass
		number in the input. 2) Check that each output AO contains the correct data.	
05	Tests that the dsmean method works with a mix of different shaped AOs as input.	Test that the dsmean method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the dismean method properly applies history.	Test that the result of applying the dsmean method can be pro- cessed back to an m-file.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'dsmean'. 2) Check that the re-built object is the same object as 'out'.	pass
07	Tests that the dsmean method can modify the input AO.	Test that the dsmean method can modify the input AO by call- ing with no output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/dsmean			
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		is dsmean(at1).	
08	Tests that the dsmean method	Test that the dsmean method	pass
	keeps the data shape of the	keeps the data shape of the in-	
	input object.	put object. The input AO must	
		be an AO with row data and an	
		AO with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
00	Check that the dsmean method	Call the method with a list of	pass
03	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/dsmean]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	
10	Check that the errors are	Call the method with a list of	pass
12	cleared for this method.	output variables and with a sin-	
		gle output variable. Additionaly	
		check that the rebuild method	
		works on the output.	
		1) Check that the output has no	pass
		error fields	

Table 37: Unit tests for ao/dsmean.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/eig			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the eig method works	Test that the eig method works	pass
02	with a vector of AOs as input.	for a vector of AOs as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'atvec' 2) Check that each output	
		AO contains the correct data.	
03	Tests that the eig method works	Test that the eig method works	pass
	with a matrix of AOs as input.	for a matrix of AOs as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as	
		in 'atmat' 2) Check that each	
		output AO contains the correct	
		data.	
04	Tests that the eig method works	Test that the eig method works	pass
	with a list of AOs as input.	for a list of AOs as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
05	Tests that the eig method works	Test that the eig method works	pass
	with a mix of different shaped	with an input of matrices and	
	AOs as input.	vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
06	Tests that the eig method	Test that the result of applying	pass
	properly applies history.	the eig method can be processed	
		back to an m-file.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to eig. 2) Check that the re-	
		built object is the same object as	
		out'.	
07	Tests that the eig method can	Test that the eig method can	pass
	modify the input AO.	modify the input AO by call-	
		ing with no output and that the	
		method doesn't change the input	
		or the function notation (with a	
		$\begin{array}{c} \text{equal sign}. \\ \hline 1 \text{ (ln sls th st 2st 4)} \\ \hline \end{array}$	
		1) Uneck that 'at4' and 'ain' are	pass
		now different. 2) Uneck that 'ain' $i = i = (-4.4)$	
		is eig(at4).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/eig			
08	Control the method with a plist.	Test that the eig method can modify the single axis controlled	pass
		by the plist and the resuld can be processed back to an m-file.	
		1) Check that the svd method applies with different options 2)	pass
		Check that the re-built objects	
		are the same object as ' $out[12]$ '.	
09	Control the method with a plist.	Test that the eig method keeps the data shape of the input ob-	pass
		ject. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
10	Check that the eig method pass	Call the method with a list of	pass
10	back the output objects to a list	output variables and with a sin-	
	of output variables or to a single	gle output variable. Additionaly	
	variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/eig] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gie output variable. Additionaly	
	single variable.	cneck that the rebuild method	
		works on the output.	
		1) Uneck that the output con-	pass
		tains the same plotinto plist	

Table 38: Unit tests for ao/eig.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/eq			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the eq method works with a vector of AOs as input.	Test that the eq method works for a vector of AOs as input. Test the positive and the nega- tive case. 1) Check the output of the eq	pass
03	Tests that the eq method works with a matrix of AOs as input.	function. Test that the eq method works for a matrix of AOs as input. Test the positive and the nega- tive case.	pass
		1) Check the output of the eq function.	pass
04	Tests that the eq method works with a list of AOs as input.	The eq method doesn't works for a list of AOs as input. Nothing to do.	pass
			pass
05	Tests that the eq method works with a mix of different shaped AOs as input.	The eq method doesn't works for a list of AOs as input. Nothing to do.	pass
			pass
06	Tests that the eq method properly applies history.	The eq method doesn't change the AO, thus will no history added. Nothing to do	pass
			pass
07	Test the eq method with an exception list. With the LTPDA toolbox 2.0 it is only possible to test the exception list with properties where a public set method exist.	Test the eq method with the exception 'name'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because aa is created at an other time. 1) Check the output.	pass
	Test the eq method with an	Test the eq method with the ev-	nass
08	exception list. With the LTPDA toolbox 2.0 it is only possible to test the exception list with properties where a public set method exist.	ception 'plotinfo'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because aa is created at an other time. 1) Check the output.	pass


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/eq			
00	Test the eq method with an	Test the eq method with the ex-	pass
09	exception list. With the LTPDA	ception 't0'. Use the option 'in-	
	toolbox 2.0 it is only possible to	ternal' to suppress the history. It	
	test the exception list with	is necessary to add 'created' to	
	properties where a public set	the exception list because as is	
	method exist.	created at an other time.	
		1) Check the output.	pass
	Test the eq method with an	Test the eq method with the ex-	pass
10	exception list. With the LTPDA	ception 'x' Use the option 'inter-	Pass
	toolbox 2.0 it is only possible to	nal' to suppress the history. It is	
	toolbox 2.0 it is only possible to	nar to suppress the instory. It is	
	proportion where a public set	evention list because as is an	
	properties where a public set	exception list because aa is cre-	
	method exist.	ated at an other time.	
		1) Check the output.	pass
11	Test the eq method with an	Test the eq method with the ex-	pass
	exception list. With the LTPDA	ception 'y'. Use the option 'inter-	
	toolbox 2.0 it is only possible to	nal' to suppress the history. It is	
	test the exception list with	necessary to add 'created' to the	
	properties where a public set	exception list because aa is cre-	
	method exist.	ated at an other time.	
		1) Check the output.	pass
10	Test the eq method with an	Test the eq method with the ex-	pass
12	exception list. With the LTPDA	ception 'xunits'. Use the option	-
	toolbox 2.0 it is only possible to	'internal' to suppress the history.	
	test the exception list with	It is necessary to add 'created' to	
	properties where a public set	the exception list because aa is	
	method exist.	created at an other time.	
		1) Check the output.	pass
	Test the eq method with an	Test the eq method with the ex-	pass
13	exception list. With the LTPDA	ception 'vunits' Use the option	Pass
	toolbox 2.0 it is only possible to	'internal' to suppress the history	
	test the exception list with	It is necessary to add 'created' to	
	properties where a public set	the exception list because an is	
	mothod evist	arouted at an other time	
	method exist.	1) Check the output	naga
	Test the as mothed with an	Toget the ag mothed with the	pass
14	rest the eq method with an	rest the eq method with the ex-	pass
	exception list. With the LIPDA	ception is. Use the option in-	
	toolbox 2.0 it is only possible to	ternal to suppress the history. It	
	test the exception list with	is necessary to add 'created' to	
	properties where a public set	the exception list because aa is	
	method exist.	created at an other time.	
		1) Check the output.	pass
15	Test the eq method with an	Test the eq method with the ex-	pass
10	exception list. With the LTPDA	ception 'description'. Use the op-	
	toolbox 2.0 it is only possible to	tion 'internal' to suppress the	
	test the exception list with	history. It is necessary to add	
	properties where a public set	'created' to the exception list be-	
	method exist.	cause as is created at an other	
		time.	
		1) Check the output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/eq			
16	Test the eq method with an	Test that the eq method uses the	pass
10	exception list which is in a plist.	exception list in a plist.	
		1) Check the output.	pass

Table 39: Unit tests for ao/eq.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/exp			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all	pass
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the [ao/exp] method	Test that the [ao/exp] method	pass
	works with a vector of objects	works for a vector of objects as	
	as input.	1) Check that the number of el-	pass
		ements in 'out' is the same as in	1
		'vec' 2) Check that each output	
		object contains the correct data.	
03	Tests that the [ao/exp] method	Test that the [ao/exp] method	pass
	as input	input	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	I man
		'mat' 2) Check that each output	
		object contains the correct data.	
04	Tests that the [ao/exp] method	Test that the [ao/exp] method	pass
	works with a list of objects as	works for a list of objects as in-	
	input.	1) Check that the number of al	nagg
		ements in 'out' is the same as in	pass
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/exp] method	Tests that the [ao/exp] method	pass
	works with a mix of different	works with a mix of different ar-	
	arrays of objects as input.	rays of objects as input.	
		1) Check that the number of el-	pass
		'mat' 2) Check that each output	
		object contains the correct data.	
0.0	Tests that the [ao/exp] method	Test that the result of applying	pass
06	properly applies history.	the [ao/exp] method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		re-built object is the same object	
		as the input.	
07	Tests that the [ao/exp] method	Test that the [ao/exp] method	pass
07	can modify the input AO.	can modify the input object by	1
		calling with no output and that	
		the method doesn't change the	
		input of the function notation	
		(with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/exp			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the [ao/exp]	
		value of the copy 4) Check that	
		out and amodi are the same	
	Test that the [ao/exp] method	Test that the [ao/exp] method	pass
08	uses the plist to get the axis.	uses the plist to get the axis.	-
		1) Check that the [ao/exp]	pass
		method applies to the x-axis 2)	-
		Check that the [ao/exp] method	
		applies to the y-axis 3) Check	
		that the [ao/exp] method applies	
		to both axes 4) Check that the	
		re-built object is the same as in	
		'out[13]'.	
00	Test the shape of the data in	Test that the [ao/exp] method	pass
09	AOs.	keeps the data shape of the in-	
		put object. The input AO data	
		must be an array with row data	
		and/or column data.	
		1) Check that the shape of the	pass
		data doesn't change.	
10	Check that the [ao/exp] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/exp] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 40: Unit tests for a o/exp.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/export			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the export method	Tests that the export method	pass
02	works with non complex data in	works with non complex data in	
	the AO.	the AO.	
		1) Check that the file exist. 2)	pass
		Check that the read data is the	
		same as the saved data.	
0.2	Tests that the export method	Tests that the export method	pass
05	works with complex data in the	works with complex data in the	
	AO.	AO.	
		1) Check that the file exist. 2)	pass
		Check that the read data is the	
		same as the saved data.	
0.4	Tests that the export method	Tests that the export method	pass
04	works with a plist which	works with a plist which contains	
	contains the filename.	the filename.	
		1) Check that the file exist. 2)	pass
		Check that the read data is the	
		same as the saved data.	

Table 41: Unit tests for ao/export.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/fft			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the [ao/fft] method	Test that the [ao/fft] method	pass
02	works with a vector of objects	works for a vector of objects as	
	as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'vec' 2) Check that each output	
		object contains the correct data.	
03	Tests that the [ao/fft] method	Test that the [ao/fft] method	pass
	works with a matrix of objects	works for a matrix of objects as	
	as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
04	Tests that the [ao/fft] method	Test that the [ao/fft] method	pass
04	works with a list of objects as	works for a list of objects as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/fft] method	Tests that the [ao/fft] method	pass
00	works with a mix of different	works with a mix of different ar-	
	arrays of objects as input.	rays of objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
06	Tests that the [ao/fft] method	Test that the result of applying	pass
	properly applies history.	the [ao/fft] method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[ao/fft]'. 2) Check that the re-	
		built object is the same object as	
		the input.	
07	Tests that the [ao/fft] method	Test that the [ao/fft] method can	pass
	can modify the input AO.	modify the input object by call-	
		ing with no output and that the	
		method doesn't change the input	
		of the function notation (with a	
		equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/fft			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that	
		the modified input is the [ao/fft]	
		value of the copy 4) Check that	
		out and amodi are the same	
00	Test the shape of the data in	Test that the [ao/fft] method	pass
09	AOs.	keeps the data shape of the in-	_
		put object. The input AO data	
		must be an array with row data	
		and/or column data.	
		1) Check that the shape of the	pass
		data doesn't change.	Pass
	Check that the [ao/fft] method	Call the method with a list of	pass
10	pass back the output objects to	output variables and with a sin-	I
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	-
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/fft] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	_
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	-
10	Tests that the fft method agrees	Test that the applying fft works	pass
12	with MATLAB's fft when	on a single AO.	
	configured to use the same	1) Check that output agrees with	pass
	parameters.	the output of MATLAB's fft.	
19	Tests that the fft method works	Test that the applying fft works	pass
10	with different data types. The	on cdata and xydata.	
	testing of tsdata types are done	1) Check that each output AO	pass
	before.	contains the correct data.	
14	Tests that the fft method works	Test that the applying fft works	pass
14	with a plist which constains the	with a plist.	
	key/value pair 'type'/'two'.	1) Check that each output AO	pass
		contains the correct data. 2)	
		Check that the re-built object is	
		the same object as 'out'.	

Table 42: Unit tests for ao/fft.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/fftfilt			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re-	pass
		turned an minto object in all	
		cases. 2) Check that all plists	
	Tests that the fftfilt method	Test that the fftfilt method works	nass
02	works with a vector of AOs as	for a vector of AOs as input.	pass
	input.	1) Check that the number of ele-	pass
	1	ments in 'out' is the same of the	1
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
03	Tests that the fftfilt method	Test that the fftfilt method works	pass
	works with a matrix of AOs as	for a matrix of AOs as input.	
	input.	1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		that each output AO contains	
		that each output AO contains	
	Tests that the fftfilt method	Test that the fftfilt method works	pass
04	works with a list of AOs as	for a list of AOs as input.	pass
	input.	1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
05	Tests that the fftfilt method	Test that the fftfilt method works	pass
	works with a mix of different	with an input of matrices and	
	shaped AOs as input.	vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in input 2) Check that each output	
		$\Delta \Omega$ contains the correct data	
	Tests that the fftfilt method	Test that the result of applying	nase
06	properly applies history	the fftfilt method can be pro-	pass
	property applies motory.	cessed back to an m-file.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'fftfilt'. 2) Check that the re-	
		built object is the same object as	
		'out'.	
07	Tests that the fftfilt method can	Test that the fftfilt method can	pass
	modify the input AO.	modify the input AO by calling	
		with no output.	
		1) Check that 'at2' and 'ain' are	pass
		now different. 2) Check that 'ain' is fitfilt($a \pm 2$)	
		is inthit(at2).	1



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/fftfilt			
08	Tests that the fftfilt method	Test that the fftfilt method keeps	pass
00	keeps the data shape of the	the data shape of the input ob-	
	input object.	ject. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
00	Check that the fftfilt method	Call the method with a list of	pass
09	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
	_	works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 43: Unit tests for a o/fftfilt.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/filtSubtract			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the filtSubtract	Test that the filtSubtract	pass
02	method works with a vector of	method works for a vector of	
	AOs as input.	AOs as input.	
		1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
0.9	Tests that the filtSubtract	Test that the filtSubtract	pass
03	method works with a matrix of	method works for a matrix of	
	AOs as input.	AOs as input.	
	_	1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
0.1	Tests that the filtSubtract	Test that the filtSubtract	pass
04	method works with a list of AOs	method works for a list of AOs	
	as input.	as input.	
	-	1) Check that the number of ele-	pass
		ments in 'out' is the square of the	-
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
	Tests that the filtSubtract	Test that the filtSubtract	pass
05	method works with a mix of	method works with an input of	1
	different shaped AOs as input.	matrices and vectors and single	
	DOES NOT APPLY TO THIS	AOs.	
	METHOD	1) Check that the number of el-	pass
		ements in 'out' is the same as in	-
		input. 2) Check that each output	
		AO contains the correct data.	
0.0	Tests that the filtSubtract	Test that the result of applying	pass
00	method properly applies history.	the filtSubtract method can be	
		processed back to an m-file.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	-
		to 'filtSubtract'. 2) Check that	
		the re-built object is the same	
		object as 'out'	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/filtSubtract			
07	Tests that the filtSubtract	Test that the tfe method can	pass
07	method can not modify the	not modify the input AO. The	
	input AO.	method must throw an error for	
		the modifier call.	
		1) Nothing to check.	pass

Table 44: Unit tests for ao/filtSubtract.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/filter			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the filter method	Test that the filter method works	pass
02	works with a vector of AOs as	for a vector of AOs as input.	
	input.	1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data. 3) Check the	
		output-filter	
0.2	Tests that the filter method	Test that the filter method works	pass
05	works with a matrix of AOs as	for a matrix of AOs as input.	
	input.	1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data. 3) Check the	
		output-filter	
04	Tests that the filter method	Test that the filter method works	pass
04	works with a list of AOs as	for a list of AOs as input.	
	input.	1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
05	Tests that the filter method	Test that the filter method works	pass
00	works with a mix of different	with an input of matrices and	
	shaped AOs as input.	vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
06	Tests that the filter method	Test that the result of apply-	pass
	properly applies history.	ing the filter method can be pro-	
		cessed back to an m-file.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'filter'. 2) Check that the re-	
		built object is the same object as	
		'out'.	
07	lests that the filter method can	1 Test that the filter method can	pass
	modify the input AO.	modify the input AO by calling	
		with no output.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/filter			
		1) Check that 'at1' and 'ain' are now different. 2) Check that 'ain' is filter(at1).	pass
08	Tests that the filter method keeps the data shape of the input object.	Test that the filter method keeps the data shape of the input ob- ject. The input AO must be an AO with row data and an AO with column data.	pass
		1) Check that the shpe of the data doesn't change.	pass
09	Tests that the filter method with a iir filter which have a different sample rate as the input object.	Test that the filter method can change the sample rate of the fil- ter and that the filter-object is returned.	pass
		1) Check that the output AO contains the correct data. 2) Check that the second output is a iir filter.	pass
10	Tests that the filter method with a fir filter.	Test that the filter method with a fir filter.	pass
		1) Check that the output AO contains the correct data. 2) Check that the second output is a fir filter.	pass
11	Check that the [ao/filter] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the same plotinfo plist	pass
12	Tests the filter method with an AO and fsdata-object and an iir filter.	Test that the result of the filter method is the product of the AO and the response of the filter.	pass
		1) Check that the output AO contains the correct data. 2) Check that the second output is a iir filter. 3) Check the units	pass
13	Tests that the filter method works with a bank of parallel filters.	Test that the filter method works for bank of parallel filters.	pass
	muers.	The correct data of the number of ele- ments in 'out' is the same of the number of elements in 'input' 2) Check that output AO contains the correct number of data. 3) Check that output AO contains the correct data. 4) Check that historic is properly assigned	pass
14	Tests that the filter method works with a bank of serial filters.	Test that the filter method works for bank of serial filters.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/filter			
	Tosts the filter method with an	1) Check that the number of ele- ments in 'out' is the same of the number of elements in 'input' 2) Check that output AO contains the correct number of data. 3) Check that output AO contains the correct data. 4) Check that histout is properly assigned	pass
15	AO of fsdata-object and a bank of iir parallel filter.	method is the product of the AO and the response of the parallel filter bank.	pass
		1) Check that the output AO contains the correct data. 2) Check that the second output is a iir filter. 3) Check the units	pass
16	Tests the filter method with an AO of fsdata-object and a bank of iir serial filter.	Test that the result of the filter method is the product of the AO and the response of the serial fil- ter bank. 1) Check that the output AO	pass
		contains the correct data. 2) Check that the second output is a iir filter. 3) Check the units	pass
17	Tests the filter method with an AO and fsdata-object and an mfir filter.	Test that the result of the filter method is the product of the AO and the response of the filter.	pass
		1) Check that the output AO contains the correct data. 2) Check that the second output is a fir filter. 3) Check the units	pass
18	Test that the filter method works with a filter embedded into a matrix and input in a	Test that the filter method works with a filter embedded into a ma- trix	pass
	plist	1) Check that the number of ele- ments in 'out' is the same of the number in the input. 2) Check that each output AO contains the correct data. 3) Check the output-filter	pass
19	Test that the filter method works with a filter embedded into a matrix and input as a	Test that the filter method works with a filter embedded into a ma- trix	pass
	second input	1) Check that the number of ele- ments in 'out' is the same of the number in the input. 2) Check that each output AO contains the correct data. 3) Check the output-filter	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/filter			
20	Test that the filter method does	Test that the filter method does	pass
20	not works with a N-dim matrix	not works with a N-dim matrix	
	of filters input in a plist.	of filters input in a plist.	
		1) Nothing to do.	pass
01	Test that the filter method does	Test that the filter method does	pass
21	not works with a N-dim matrix	not works with a N-dim matrix	
	of filters input as a second	of filters input in a plist.	
	object	1) Nothing to do.	pass

Table 45: Unit tests for a o/filter.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/filtfilt			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the filtfilt method works with a vector of AOs as input.	Test that the filtfilt method works for a vector of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data. 3) Check the output-filter	pass
03	Tests that the filtfilt method works with a matrix of AOs as input.	Test that the filtfilt method works for a matrix of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data. 3) Check the output-filter	pass
04	Tests that the filtfilt method works with a list of AOs as	Test that the filtfilt method works for a list of AOs as input.	pass
	mput.	1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
05	Tests that the filtfilt method works with a mix of different shaped AOs as input.	Test that the filtfilt method works with an input of matrices and vectors and single AOs. 1) Check that the number of el- ements in 'out' is the same as in input 2) Check that each output	pass pass
06	Tests that the filtfilt method properly applies history.	AO contains the correct data. Test that the result of applying the filtfilt method can be pro-	pass
		 cessed back to an m-file. 1) Check that the last entry in the history of 'out' corresponds to 'filtfilt'. 2) Check that the rebuilt object is the same object as 'out'. 	pass
07	Tests that the filtfilt method can modify the input AO.	Test that the filtfilt method can modify the input AO by calling with no output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/filtfilt			
		1) Check that 'at1' and 'ain' are now different. 2) Check that 'ain' is filtfilt(at1).	pass
08	Tests that the filtfilt method keeps the data shape of the input object.	Test that the filtfilt method keeps the data shape of the in- put object. The input AO must be an AO with row data and an AO with column data.	pass
		1) Check that the shpe of the data doesn't change.	pass
09	Tests that the filtfilt method with a iir filter which have a different sample rate as the input object.	Test that the filtfilt method can change the sample rate of the fil- ter and that the filtfilt-object is returned.	pass
		1) Check that the output AO contains the correct data. 2) Check that the second output is a iir filtfilt.	pass
10	Tests that the filtfilt method with a fir filter.	Test that the filtfilt method with a fir filter.	pass
		1) Check that the output AO contains the correct data. 2) Check that the second output is a fir filter.	pass
11	Check that the [ao/filtfilt] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the same plotinfo plist	pass
12	Tests the filtfilt method with an AO and fsdata-object and an iir filter.	Test that the result of the filtfilt method is the product of the AO and the response of the filter.	pass
		1) Check that the output AO contains the correct data. 2) Check that the second output is a iir filter. 3) Check the units	pass
13	Tests the filtfilt method with an AO and fsdata-object and an iir filter.	Test that the result of the filtfilt method is the product of the AO and the response of the filter.	pass
		1) Check that the output AO contains the correct data. 2) Check that the second output is a fir filter. 3) Check the units	pass

Table 46: Unit tests for ao/filtfilt.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/find			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the find method works with a vector of AOs as input.	Test that the find method works for a vector of AOs as input.1) Check that the number of el- ements in 'out' is the same as in 'atvec' 2) Check that each output	pass pass
03	Tests that the find method works with a matrix of AOs as input.	 AO contains the correct data. Tests that the find method works with a matrix of AOs as input. 1) Check that the number of elements in 'out' is the same as 	pass pass
	Tosts that the find method	in 'atmat' 2) Check that each output AO contains the correct data.	Dage
04	works with a list of AOs as input.	 rests that the find method works with a list of AOs as input. 1) Check that the number of el- ements in 'out' is the same as in 'atmat' 2) Check that each output AO contains the correct data. 	pass
05	Tests that the find method works with a mix of different shaped AOs as input.	 Tests that the find method works with a mix of different shaped AOs as input. 1) Check that the number of el- ements in 'out' is the same as in 'atmat' 2) Check that each output AO contains the correct data. 	pass
06	Tests that the find method properly applies history.	 Test that the result of applying the find method can be processed back. 1) Check that the last entry in the history of 'out' corresponds to 'finf'. 2) Check that the rebuilt object is the same object as 	pass
07	Tests that the find method can modify the input AO.	the input. Test that the find method can modify the input AO by call- ing with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/find			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the abs value of	
		the copy 4) Check that out and	
		amodi are the same	
	Test that the find method works	Test that the find method works	pass
08	with different queries to the	with different queries to the x-/y-	-
	x-/y- axis.	axis.	
		1) Check the output	pass
		Test that the find method keeps	pass
09	Test the shape of the output.	the data shape of the input ob-	1
		iect. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shape of the	pass
		data doesn't change.	1
10	Check that the find method	Call the method with a list of	pass
10	pass back the output objects to	output variables and with a sin-	1
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	P
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		iect as 'out'	
	Check that the [ao/find] method	Call the method with a list of	nass
11	pass back the output objects to	output variables and with a sin-	Pass
	a list of output variables or to a	gle output variable Additionaly	
	single variable	check that the rebuild method	
	single variable.	works on the output	
		1) Check that the output con-	nass
		tains the same plotinfo plist	Pass
	Test that the find method works	Test that the find method works	pass
12	with AOs which have cdata	with AOs which have cdata	Pass
		1) Check the output	pass
	Test that the find method works	Test that the find method works	pass
13	with a plist which contains	with a plist which contains dif-	r the
	different queries to the x-/y-	ferent queries to the x/y - axis	
	axis	1) Check the output	nass
	anio.	1) Oncer the output	pass

Table 47: Unit tests for ao/find.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/firwhiten			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters	pass
02	Tests that the firwhiten method works with a vector of AOs as	Test that the firwhiten method works for a vector of AOs as in-	pass
	input.	put. 1) Check the number of elements in 'out' 2) Check the number of filters (outf) and noise-floor es- timates (outxx) 3) Check that each output AO contains the cor-	pass
03	Tests that the firwhiten method works with a matrix of AOs as input.	Test that the firwhiten method works for a matrix of AOs as in- put.	pass
		1) Check the number of elements in 'out' 2) Check the number of filters (outf) and noise-floor es- timates (outxx) 3) Check that each output AO contains the cor- rect data.	pass
04	Tests that the firwhiten method works with a list of AOs as input.	Test that the firwhiten method works for a list of AOs as input. 1) Check the number of elements in 'out' 2) Check the number of filters (outf) and noise-floor es- timates (outxx) 3) Check that each output AO contains the cor- rect data.	pass pass
05	Tests that the firwhiten method works with a mix of different shaped AOs as input.	Test that the firwhiten method works with an input of matrices and vectors and single AOs.	pass
		1) Check the number of elements in 'out' 2) Check the number of filters (outf) and noise-floor es- timates (outxx) 3) Check that each output AO contains the cor- rect data.	pass
06	Tests that the firwhiten method properly applies history.	Test that the result of applying the firwhiten method can be pro- cessed back to an m-file.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'firwhiten'. 2) Check that the re-built object is the same object as 'out'.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/firwhiten			
07	Tests that the firwhiten method	Test that the firwhiten method	pass
07	can modify the input AO.	can modify the input AO by call-	
		ing with no output.	
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		is firwhiten(at1).	
00	Tests that the firwhiten method	Test that the firwhiten method	pass
08	keeps the data shape of the	keeps the data shape of the in-	
	input object.	put object. The input AO must	
	1 0	be an AO with row data and an	
		AO with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	1
	Tests that the firwhiten method	Test that the result of apply-	pass
09	with a complex plist.	ing the firwhiten method with a	1
	1 1	complex plist can be processed	
		back to a m-file.	
		1) Check the output data 2)	pass
		Check the output filter 3) Check	T
		the noise-floor estimates 4)	
		Check that the re-built object is	
		the same object as 'out'.	
	Test the spectral falttening	Test that the application of the	pass
10	capability of firwhiten method.	firwhiten method enhances the	T
		spectral flatness of input data.	
		1) Calculate PSD of input and	pass
		whitened data 2) Compare rela-	1
		tive spectral flatness coefficients	
	Check that the [ao/firwhiten]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	1
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	-
10	Check that the errors are	Call the method with a list of	pass
12	cleared for this method.	output variables and with a sin-	.
		gle output variable. Additionaly	
		check that the rebuild method	
		works on the output.	
		1) Check that the output has no	pass
		error fields	1

Table 48: Unit tests for ao/firwhiten.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/fixfs			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually	pass
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the fixts method	Test that the fixts method works	pass
	works with a vector of AOs as	tor a vector of AOs as input.	naga
	input.	n) Check that the number of ele-	pass
		number in the input 2) Check	
		that each output AO contains	
		the correct data.	
<u></u>	Tests that the fixfs method	Test that the fixfs method works	pass
03	works with a matrix of AOs as	for a matrix of AOs as input.	-
	input.	1) Check that the number of ele-	pass
		ments in 'out' is the same as the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
04	Tests that the fixts method	Test that the fixts method works	pass
	works with a list of AOs as	tor a list of AOs as input.	Daga
	input.	n) Check that the number of ele-	pass
		number in the input 2) Check	
		that each output AO contains	
		the correct data.	
	Tests that the fixfs method	Test that the fixfs method works	pass
05	works with a mix of different	with an input of matrices and	-
	shaped AOs as input.	vectors and single AOs. Addi-	
		tionally define a 'fs' in a plist.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data. 3)	
		Check that 'tu' and 'Is' are cor-	
	Tests that the fixfa method	Test that the result of apply	Dago
06	properly applies history	ing the fixfs method can be pro-	pass
	property applies instory.	cessed back to an m-file Addi-	
		tionally define a 'fs' in a plist.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	.
		to 'fixfs'. 2) Check that the re-	
		built object is the same object as	
		'out'.	
07	Tests that the fixfs method can	Test that the fixfs method can	pass
~ .	modify the input AO.	modify the input AO by calling	
		with no output.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/fixfs			
		1) Check that 'at1' and 'ain' are now different. 2) Check that 'ain' is fixfs(at1).	pass
08	Tests that the fixfs method keeps the data shape of the input object.	Test that the fixfs method keeps the data shape of the input ob- ject. The input AO must be an AO with row data and an AO with column data.	pass
		1) Check that the shpe of the data doesn't change.	pass
09	Check that the fixfs method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
10	Tests that the fixfs method works with a list of AOs as input and different 't0' and 'fs'	Test that the fixfs method works for a list of AOs as input and dif- ferent 't0' and 'fs'	pass
	for the inputs.	1) Check that the number of ele- ments in 'out' is the same as the number in the input. 2) Check that each output AO contains the correct data. 3) Check that each output contains the correct frequency and start time.	pass
11	Check that the [ao/fixfs] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
	Tests that the fixfs method	tains the same plotinfo plist	pass
12	works with the method 'samples'.	for the method 'samples'. 1) Check that each output AO contains the correct data. 2) Check that each output contains the correct frequency and start time.	pass
13	Tests that the fixfs method works with antialising filter.	Test that the fixfs method works for the antialising filters iir and fir.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/fixfs			
		1) Check that each output AO	pass
		contains the correct data. 2)	
		Check that each output contains	
		the correct frequency and start	
		time.	
14	Tests that the fixfs method	Test that the fixfs method works	pass
14	works with an AO with non	for an AO with non evenly sam-	
	evenly sampled data.	ples data.	
		1) Check that each output AO	pass
		contains the correct data. 2)	
		Check that each output contains	
		the correct frequency and start	
		time.	

Table 49: Unit tests for ao/fixfs.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/fs			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the fs method works with a vector of AOs as input.	The fs method doesn't work with a vector of AOs. Nothing to do	pass
			pass
03	Tests that the fs method works with a matrix of AOs as input.	The fs method doesn't work with a matrix of AOs. Nothing to do	pass
			pass
04	Tests that the fs method works with a list of AOs as input.	The fs method doesn't work with a list of AOs. Nothing to do	pass
			pass
05	Tests that the fs method works with a mix of different shaped AOs as input.	The fs method can only return the fs value of one AO. Nothing to do	pass
			pass
06	Tests that the fs method properly applies history.	The fs method doesn't change the AO, thus will no history added. Nothing to do	pass
			pass
07	Tests that the fs method works for AOs with different data objects.	Test that the fs method returns the fs value for AOs with cdata, fsdata, tsdata and xydata ob- jects. 1) Check the output.	pass

Table 50: Unit tests for a o/fs.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/ge			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.9	Tests that the ge method	Test that the ge method works	pass
02	compare an AO with scalar	with relational operators and the	
	value	function command. Use for this	
		AOs with different data objects.	
		1) Check that the result of	pass
		the 'relational operator' and	
		the 'function' command are the	
		same. 2) Check that each output	
		AO contains the correct data.	
02	Tests that the ge method	Test that the ge method works	pass
05	compare an AO with one other	with relational operators and the	
	AO	function command. Use for this	
		AOs with different data objects.	
		Remark that both AOs must	
		have the same size.	
		1) Check that the result of	pass
		the 'relational operator' and	
		the 'function' command are the	
		same. 2) Check that each output	
		AO contains the correct data.	

Table 51: Unit tests for a o/ge.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/get			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests the get method of the ao	Test that the get returns returns	pass
02	class.	the value of the specified prop-	
		erty. Do this for all properties of	
		the AO.	
		1) Check the correct value of the	pass
		output	
03	Tests that the get method works	Test that the get returns returns	pass
00	with a plist.	the value of the specified prop-	
		erty which is defined in a plist.	
		1) Check the correct value of the	pass
		output	
04	Tests the get method of the ao	Test that the get throws an error	pass
04	class.	if the input are more than one	
		AO.	
		1) Nothing to test	pass

Table 52: Unit tests for ao/get.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/gt			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.0	Tests that the gt method	Test that the gt method works	pass
02	compare an AO with scalar	with relational operators and the	
	value	function command. Use for this	
		AOs with different data objects.	
		1) Check that the result of	pass
		the 'relational operator' and	
		the 'function' command are the	
		same. 2) Check that each output	
		AO contains the correct data.	
0.2	Tests that the gt method	Test that the gt method works	pass
05	compare an AO with one other	with relational operators and the	
	AO	function command. Use for this	
		AOs with different data objects.	
		Remark that both AOs must	
		have the same size.	
		1) Check that the result of	pass
		the 'relational operator' and	
		the 'function' command are the	
		same. 2) Check that each output	
		AO contains the correct data.	

Table 53: Unit tests for ao/gt.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/heterodyne			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the heterodyne method works with a vector of AOs as input.	Test that the heterodyne method works for a vector of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
03	Tests that the heterodyne method works with a matrix of AOs as input.	Test that the heterodyne method works for a matrix of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
04	Tests that the heterodyne method works with a list of AOs	Test that the heterodyne method works for a list of AOs as input.	pass
	as input.	1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
05	Tests that the heterodyne method works with a mix of different shaped AOs as input.	Test that the heterodyne method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the heterodyne method properly applies history.	Test that the result of applying the heterodyne method can be processed back to an m-file.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'heterodyne'. 2) Check that the re-built object is the same object as 'out'.	pass
07	Tests the heterodyne method functionality.	Build reference signal, mixed sig- nal and heterodyne the 2nd to obtain the first Downsample is set to 'no'	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/heterodyne			
		Test that we can recover the ini-	pass
		tial signal after heterodyne up to	
		a numerical error given by tol	
08	Tests the heterodyne method	Build reference signal, mixed sig-	pass
00	functionality.	nal and heterodyne the 2nd to	
		obtain the first Downsample is	
		set to 'yes'	
		Test that we can recover the ini-	pass
		tial signal after heterodyne up to	
		a numerical error given by tol	
11	Check that the [ao/heterodyne]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 54: Unit tests for a o/heterodyne.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/hist			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set	pass
		individually.	
		1) Check that getInfo call re-	pass
		turned an minto object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the hist method	Test that the hist method works	pass
	works with a vector of AOs as	for a vector of AOs as input.	
	input.	1) Check that the number of el-	pass
		ements in out is the same as in	
		AQ contains the compatidate	
	Tests that the hist method	Togt that the higt method works	naga
03	works with a matrix of AOs as	for a matrix of AOg ag input	pass
	input	1) Chock that the number of al	nase
	input.	omonts in 'out' is the same as	pass
		in 'stmat' 2) Check that each	
		α output AO contains the correct	
		data	
	Tests that the hist method	Test that the hist method works	nass
04	works with a list of AOs as	for a list of AOs as input	pass
	input.	1) Check that the number of el-	pass
	mpau	ements in 'out' is the same as in	Pass
		input. 2) Check that each output	
		AO contains the correct data.	
05	Tests that the hist method	Test that the hist method works	pass
05	works with a mix of different	with an input of matrices and	-
	shaped AOs as input.	vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
06	Tests that the hist method	Test that the result of applying	pass
00	properly applies history.	the hist method can be processed	
		back to an m-file.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'hist'. 2) Check that the re-	
		built object is the same object as	
		Yout'.	
07	Tests that the hist method can	Test that the hist method can	pass
	modify the input AO.	modify the input AO by calling	
		with no output. Remark that the	
		the histogram method because	
		an AO have a property with the	
		an AO have a property with the	
		at hist returns the value in the	
		property higt	
		property mst.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/hist			
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		is hist(at1).	
08	Tests that the hist method	Test that the hist method keeps	pass
08	keeps the data shape of the	the data shape of the input ob-	
	input object.	ject. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
00	Check that the hist method pass	Call the method with a list of	pass
09	back the output objects to a list	output variables and with a sin-	
	of output variables or to a single	gle output variable. Additionaly	
	variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
10		Test the hist method with a fac-	pass
10	Control the method with a plist.	tor and an offset.	
		1) Check that the hist method	pass
		with defined number of bins 2)	
		Check that the re-built object is	
		the same object as 'out'.	
11	Check that the [ao/hist] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	
10	Control the method with a plist	Test the hist method with a fac-	pass
12	Control the method with a plist.	tor and an offset.	
		1) Check that the hist method	pass
		with set of bin centers 2) Check	
		that the re-built object is the	
		same object as 'out'.	

Table 55: Unit tests for ao/hist.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/ifft			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually	pass
		1) Check that getInfo call re-	pass
		turned an minfo object in all	pass
		cases. 2) Check that all plists	
		have the correct parameters.	
	Tests that the ifft method works	Test that the ifft method works	pass
02	with a vector of AOs as input.	for a vector of AOs as input.	
	1	1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
0.2	Tests that the ifft method works	Test that the ifft method works	pass
05	with a matrix of AOs as input.	for a matrix of AOs as input.	
		1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
04	Tests that the ifft method works	Test that the ifft method works	pass
04	with a list of AOs as input.	for a list of AOs as input.	
		1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
05	Tests that the ifft method works	Test that the ifft method works	pass
	with a mix of different shaped	with an input of matrices and	
	AOs as input.	vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
06	lests that the int method	lest that the result of applying	pass
	property applies instory.	back to an m file	
		1) Check that the last entry in	naga
		the history of 'out' corresponde	pass
		the listory of out corresponds	
		built object is the same object as	
		out'	
	Tests that the ifft method can	Test that the ifft method can	nase
07	modify the input $A\Omega$	modify the input $A\Omega$ by calling	pass
	mouny the input AO.	with no output	
		1) Check that 'at?' and 'ain' are	nase
		now different 2) Check that 'sin'	Pass
		is ifft(at2)	
1			1



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/ifft			
08	Tests that the ifft method keeps	Test that the ifft method keeps	pass
00	the data shape of the input	the data shape of the input ob-	
	object.	ject. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
00	Check that the ifft method pass	Call the method with a list of	pass
09	back the output objects to a list	output variables and with a sin-	
	of output variables or to a single	gle output variable. Additionaly	
	variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
10	Tests that the ifft method	Test that the applying ifft works	pass
10	agrees with MATLAB's ifft	on two AOs.	
	when configured to use the same	1) Check that output agrees with	pass
	parameters.	the output of MATLAB's ifft.	
11	Check that the [ao/ifft] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	
19	Tests that the ifft method	Test that the applying ifft works	pass
14	agrees with MATLAB's ifft	on a single AO with 'nonsymmet-	
	when configured to use the same	ric' option.	
	parameters.	1) Check that output agrees with	pass
		the output of MATLAB's ifft.	

Table 56: Unit tests for ao/ifft.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/imag			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct personnet	pass
		The table the correct parameters.	
02	Tests that the imag method	for a most on of AOs as input	pass
	works with a vector of AOs as	1) Check that the mark much al	
	mput.	1) Check that the number of el-	pass
		'entents in out is the same as in	
		AQ contains the connect data	
	Tests that the image mathed	AO contains the correct data.	-
03	rests that the imag method	for a matrix of AOa as input	pass
	works with a matrix of AOs as	1) Check that the number of al	magg
	input.	1) Check that the number of el-	pass
		in 'atmat' 2) Check that each	
		m at mat 2) Check that each output AO contains the correct	
		deta	
	Tosts that the imag method	Tost that the image method works	nase
04	works with a list of AOs as	for a list of $\Delta \Omega_{\rm S}$ as input	pass
	input	1) Check that the number of el-	nase
	mput.	ements in 'out' is the same as in	pass
		input 2) Check that each output	
		AO contains the correct data	
	Tests that the imag method	Test that the imag method works	nass
05	works with a mix of different	with an input of matrices and	pass
	shaped AOs as input	vectors and single AOs	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	P
		input. 2) Check that each output	
		AO contains the correct data.	
0.0	Tests that the imag method	Test that the result of applying	pass
06	properly applies history.	the imag method can be pro-	
		cessed back to an m-file.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'imag'. 2) Check that the re-	
		built object is the same object as	
		'out'.	
07	Tests that the imag method can	Test that the imag method can	pass
	modify the input AO.	modify the input AO by call-	
		ing with no output and that the	
		method doesn't change the input	
		of the function notation (with a	
		equal sign).	
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		is imag(at1).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/imag			
08	Control the method with a plist	Test that the imag method can	pass
08	Control the method with a plist.	modify the single axis controlled	
		by the plist and the resuld can be	
		processed back to an m-file.	
		1) Check that the imag method	pass
		applies to the x-axis 2) Check	
		that the imag method applies to	
		the y-axis 3) Check that the imag	
		method applies to both axes 4)	
		Check that the re-built objects	
		are the same object as ' $out[13]$ '.	
00	Control the method with a -list	Test that the imag method keeps	pass
09	Control the method with a plist.	the data shape of the input ob-	
		ject. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
10	Check that the imag method	Call the method with a list of	pass
10	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/imag]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 57: Unit tests for ao/imag.


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/index			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the index method works with a vector of AOs as input.	Test that the index method works for a vector of AOs as input. The following indexing should work: $I = [1 2 3]$ or (I/J) = [(1,1), (1,2), (1,3)]	pass
		1) Check that the index method selects the correct object.	pass
03	Tests that the index method works with a matrix of AOs as input.	Test that the index method works for a matrix of AOs as input. The following indexing should work: $I = [1 \ 3 \ 5]$ or (I/J) $= [(1,1), (1,2), (1,3)] [2 \ 4 \ 6]$ [(2,1), (2,2), (2,3)]	pass
		selects the correct object.	pass
04	Tests that the index method works with a list of AOs as	The index method doesn't work for a list of AOs as input.	pass
	input.	1) Nothing to test.	pass
05	rests that the index method properly applies history.	an additional history step.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'index'.	pass
06	Tests that the index method works for the modifier command.	Tests that the index method works for the modifier command. 1) Check that the history-plist contains the used indices. 2) Check that the index method se- lects the correct object	pass pass
07	Control the method with a plist.	Test that the index method can be controled with a plist.	pass
		1) Check that the history-plist contains the used indices. 2) Check that the index method se- lects the correct object	pass
08	Test that the index method selects more objects if I have more indices.	Test that the index method se- lects more objects if I have more indices.	pass
		1) Check that the history-plist contains the used indices. 2) Check that the index method se- lects the correct object	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/index			
11	Check that the [ao/index]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 58: Unit tests for a o/index.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/integrate			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the integrate method	Test that the integrate method	pass
02	works with a vector of AOs as	works for a vector of AOs as in-	
	input.	put. Use for this test the trape-	
		zoidal method.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'atvec' 2) Check that each output	
		AO contains the correct data.	
03	Tests that the integrate method	Test that the integrate method	pass
00	works with a matrix of AOs as	works for a matrix of AOs as in-	
	input.	put. Use for this test the Trape-	
		zoidal method.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as	
		in 'atmat' 2) Check that each	
		output AO contains the correct	
		data.	
04	Tests that the integrate method	Test that the integrate method	pass
04	works with a list of AOs as	works for a list of AOs as input.	
	input.	Use for this test the trapezoidal.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
05	Tests that the integrate method	Test that the integrate method	pass
00	works with a mix of	works with an input of matrices	
	integrateerent shaped AOs as	and vectors and single AOs. Use	
	input.	for this test the trapezoidal.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
06	Tests that the integrate method	Test that the result of applying	pass
	properly applies history.	the integrate method can be pro-	
		cessed back to an m-file.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'integrate'. 2) Check that the	
		re-built object is the same object	
		as 'out'.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/integrate			
07	Tests that the integrate method	Test that the integrate method	pass
01	can modify the input AO.	can modify the input AO by call-	
		ing with no output. Use for this	
		test the trapezoidal.	
		1) Check that 'at1' and 'ain'	pass
		are now integratement. 2) Check	
		that 'ain' is integrate(at1).	
00	Tests that the integrate method	Test that the integrate method	pass
00	keeps the data shape of the	keeps the data shape of the in-	
	input object.	put object. The input AO must	
		be an AO with row data and an	
		AO with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
00	Check that the integrate	Call the method with a list of	pass
09	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
10	Control the method with a plict	Test the trapezoidal.	pass
10	Control the method with a plist.	1) Check that the integrate	pass
		method uses the trapezoidal	
		method. 2) Check that the re-	
		built object is the same object as	
		'out'.	
11	Check that the [ao/integrate]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 59: Unit tests for ao/integrate.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/interp			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the interp method works with a vector of AOs as input.	Test that the interp method works for a vector of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
03	Tests that the interp method works with a matrix of AOs as input.	Test that the interp method works for a matrix of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
04	Tests that the interp method works with a list of AOs as input.	Test that the interp method works for a list of AOs as input. 1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains	pass pass
		that each output AO contains the correct data	
05	Tests that the interp method works with a mix of different shaped AOs as input.	Test that the interp method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the interp method properly applies history.	Test that the result of applying the interp method can be pro- cessed back to an m-file.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'interp'. 2) Check that the re- built object is the same as 'out'.	pass
07	Tests that the interp method can modify the input AO.	Test that the interp method can modify the input AO by calling with no output.	pass
		now different. 2) Check that 'ain' is interp(at1).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/interp			
00	Tests that the interp method	Test that the interp method	pass
08	keeps the data shape of the	keeps the data shape of the in-	
	input object.	put object. The input AO must	
		be an AO with row data and an	
		AO with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
00	Check that the interp method	Call the method with a list of	pass
09	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	-
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
10	Tests that the interp method	Test that the interp method	pass
10	can different interpolations.	can all of MATLAB interpo-	
		lates methods. 'nearest' - Nearest	
		neighbor interpolation 'linear' -	
		Linear interpolation 'spline' -	
		Cubic spline interpolation (see	
		UTPs above) 'pchip' - Piecewise	
		cubic Hermite interpolation	
		1) Check the different interpola-	pass
		tions 2) Check that the re-built	
		objects are the same as 'out13'.	
11	Check that the [ao/interp]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 60: Unit tests for ao/interp.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



01 Tests that the getInfo call works for this method. Tests that the getInfo call works for no sets, all sets, and cach set individually. pass 02 Tests that the interprinsing method works with a vector of AOs as input. Test that the interprinsing method works with a vector of AOs as input. Test that the interprinsing method works for a vector of AOs as input. Known gaps at the position idx = 30 and 51 pass 03 Tests that the interprinsing method works with a matrix of AOs as input. Test that the interprinsing method works for a matrix of AOs as input. When we correct data. Test that the interprinsing method works with a matrix of AOs as input. Known gaps at the correct data. pass 04 Tests that the interprinsing method works with a list of AOs as input. Test that the interprinsing method works with a list of AOs as input. Known gaps at the position idx = 30 and 51 pass 05 Tests that the interprinsing method works with a mix of different shaped AOs as input. Test that the interprinsing method works with a mix of different shaped AOs as input. Test that the interprinsing method works with a mix of different shaped AOs as input. Test that the interprinsing method works with a mix of different shaped AOs as input. Test that the interprinsing method works with a mix of different shaped AOs as input. Test that the interprinsing method works with a mix of different shaped AOs as input. Test that the interprinsing method works with a mix of different shaped AOs as input. Test that the interprinsing method works with a mix of different shaped AOs as input. Test that the interpretising method works with	ao/interpmissing			
01 for this method. for no sets, all sets, and each set individually. 1) Check that getInfo call returned an minfo object in all cases. 2) Check that all plists have the correct parameters. pass 02 Tests that the interpmissing method works with a vector of AOs as input. Test that the interpmissing method works for a vector of AOs as input. Known gaps at the position idx = 30 and 51 pass 03 Tests that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works for a matrix of AOs as input. Test that the interpmissing method works for a matrix of AOs as input. Known gaps at the position idx = 30 and 51 pass 04 Tests that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works with a list of AOs as input. Known gaps at the position idx = 30 and 51 pass 04 Tests that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works with a list of AOs as input. Known gaps at the position idx = 30 and 51 pass 05 Tests that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test th	01	Tests that the getInfo call works	Test that the getInfo call works	pass
02 Tests that the interpmissing method works with a vector of AOs as input. Test that the interpmissing method works with a vector of AOs as input. Test that the interpmissing method works with a vector of AOs as input. Test that the interpmissing method works with a vector of AOs as input. Known gaps at the position idx = 30 and 51 pass 03 Tests that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works with a matrix of AOs as input. Known gaps at the position idx = 30 and 51 pass 04 Tests that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works with a list of AOs as input. pass method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the nerwell of apply	01	for this method.	for no sets, all sets, and each set	
02 Tests that the interpmissing method works with a vector of AOs as input. Tests that the interpmissing method works with a vector of AOs as input. Test that the interpmissing method works for a vector of AOs as input. Known gaps at the position idx = 30 and 51 pass 03 Tests that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works for a list of AOs as input. Known gaps at the position idx = 30 and 51 pass 04 Tests that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works for a list of AOs as input. Known gaps at the position idx = 30 and 51 pass 05 Tests that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. 06			individually.	
02 Tests that the interpmissing method works with a vector of AOs as input. Test that the interpmissing method works with a vector of AOs as input. Known gaps at the position idx = 30 and 51 pass 03 Tests that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works with a matrix of AOs as input. Snown gaps at the position idx = 30 and 51 pass 03 Tests that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works with a matrix of AOs as input. Snown gaps at the position idx = 30 and 51 pass 04 Tests that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works with a list of AOs as input. Snown gaps at the position idx = 30 and 51 pass 05 Tests that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpm			1) Check that getInfo call re-	pass
02 Tests that the interpmissing method works with a vector of AOs as input. Test that the interpmissing method works for a vector of AOs as input. Test that the interpmissing method works with a vector of AOs as input. Test that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works for a vector of AOs as input. Known gaps at the position idx = 30 and 51 pass 03 Tests that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works with a matrix of AOs as input. Known gaps at the position idx = 30 and 51 pass 04 Tests that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works with a list of AOs as input. Known gaps at the position idx = 30 and 51 pass 05 Tests that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works with a list of AOs as input. Known gaps at the position idx = 30 and 51 pass 05 Tests that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input.			turned an minfo object in all	
02 Tests that the interpmissing method works with a vector of AOs as input. Test that the interpmissing method works for a vector of AOs as input. Known gaps at the position idx = 30 and 51 pass 03 Tests that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works for a vector of AOs as input. So and 51 pass 03 Tests that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works for a matrix of AOs as input. Known gaps at the position idx = 30 and 51 pass 04 Tests that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works for a list of AOs as input. pass 05 Tests that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the number of el- ments in 'out' is the same as in input. 2) Check that each output AO contains the correct data. pass 06 Tests that the interpmissing method properly applies history. Test that the result of applying the interpmissing method can be processed back to an m-file. pass			cases. 2) Check that all plists	
02 Tests that the interpmissing method works with a vector of AOs as input. Test that the interpmissing method works for a vector of AOs as input. Known gaps at the position idx = 30 and 51 pass 03 Tests that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works for a matrix of AOs as input. pass 03 Tests that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works for a matrix of AOs as input. Known gaps at the position idx = 30 and 51 pass 04 Tests that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works for a list of AOs as input. Known gaps at the position idx = 30 and 51 pass 05 Tests that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with an input of matrices and vectors and single AOs. Known gaps at th			have the correct parameters.	
02 method works with a vector of AOs as input. method works for a vector of AOs as input. Known gaps at the position idx = 30 and 51 03 Tests that the interpmissing method works with a matrix of AOs as input. Tests that the interpmissing method works with a matrix of AOs as input. Tests that the interpmissing method works with a matrix of AOs as input. Tests that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works for a matrix of AOs as input. Test that the interpmissing method works for a matrix of AOs as input. Test that the interpmissing method works for a list of AOs as input. Test that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the number of el- ments in 'out' is the same as in single AOs. Known gaps at the position id a = 30 and 51 pass 06 Tests that the interpmissing method properly applies history. Test that the result of applying the interpmissing method can be processed back to an mefile. pass	0.2	Tests that the interpmissing	Test that the interpmissing	pass
AOs as input. AOs as input. Known gaps at the position idx = 30 and 51 i) Check that the number of elements in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data. pass 03 Tests that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works for a matrix of AOs as input. The square of the number in the input. 2) Check that each output AO contains the correct data. pass 04 Tests that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works with a list of AOs as input. Chock that each output AO contains the correct data. pass 04 Tests that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works with a list of AOs as input. Known gaps at the position idx = 30 and 51 pass 05 Tests that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the number of elements in 'out' is the square of the number in the input 2) Check that each output AO contains the correct data. pass 05 Tests that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the number of elements in 'out' is the square of the number of elements in 'out' is the same	02	method works with a vector of	method works for a vector of	
03 Tests that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works for a matrix of AOs as input. Test that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works for a matrix of AOs as input. AO contains the correct data. Test that the interpmissing method works with a list of AOs as input. Known gaps at the position idx = 30 and 51 Test that the interpmissing method works for a list of AOs as input. Known gaps at the position idx = 30 and 51 04 Tests that the interpmissing method works with a list of AOs as input. Known gaps at the position idx = 30 and 51 Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with an input of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51 Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with an input of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51 Test that the interpmissing method works with an input of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51 <td></td> <td>AOs as input.</td> <td>AOs as input. Known gaps at</td> <td></td>		AOs as input.	AOs as input. Known gaps at	
1) Check that the number of elements in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.pass03Tests that the interpmissing method works with a matrix of AOs as input.Test that the interpmissing method works for a matrix of AOs as input. Known gaps at the position idx = 30 and 51pass04Tests that the interpmissing method works with a list of AOs as input.Test that the interpmissing method works for a list of AOs as input. You work for a list of AOs as input.pass04Tests that the interpmissing method works with a list of AOs as input.Test that the interpmissing method works for a list of AOs as input. Known gaps at the position idx = 30 and 51pass05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the number of elements in 'out' is the same as in input. 2) Check that the number of elements in 'out' is the same as in input. 2) Check that the number of elements in 'out' is the same as in input. 2) Check that the number of elements in 'out' is the same as in input. 2) Check that the number of elements in 'out' is the same as in input. 2) Check that the number of elements in 'out' i			the position $idx = 30$ and 51	
03Tests that the interpmissing method works with a matrix of AOs as input.Test that the interpmissing method works with a matrix of AOs as input.Test that the interpmissing method works for a matrix of AOs as input. Known gaps at the position idx = 30 and 51Test that the interpmissing method works for a matrix of AOs as input. So and 51pass04Tests that the interpmissing method works with a list of AOs as input.Test that the interpmissing method works for a list of AOs as input. Nnown gaps at the position idx = 30 and 51pass04Tests that the interpmissing method works with a list of AOs as input.Test that the interpmissing method works for a list of AOs as input. Known gaps at the position idx = 30 and 51pass05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of disigle AOs. Known gaps at the position idx = 30 and 51pass method works with a mix of disigle AOs. Known gaps at the position idx = 30 and 51pass06Tests that the interpmissing method properly applies history.Test that the result of applying the interpmissing method can be processed back to an m-file.pass			1) Check that the number of ele-	pass
03 Tests that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works for a matrix of AOs as input. Known gaps at the position idx = 30 and 51 pass 04 Tests that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works for a list of AOs as input. pass 04 Tests that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works for a list of AOs as input. Known gaps at the position idx = 30 and 51 pass 05 Tests that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the number of elements in 'out' is the same as in input. 2) Check that each output AO contains the correct data. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the number of elements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.<			ments in 'out' is the square of the	
03Tests that the interpmissing method works with a matrix of AOs as input.Test that the interpmissing method works for a matrix of AOs as input. Known gaps at the position idx = 30 and 51pass method works for a matrix of AOs as input. Known gaps at the position idx = 30 and 5104Tests that the interpmissing method works with a list of AOs as input.Test that the interpmissing method works for a list of AOs as input.Test that the interpmissing method works for a list of AOs as input.Test that the interpmissing method works for a list of AOs as input.Test that the interpmissing method works for a list of AOs as input.Test that the interpmissing method works with a list of AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of dising AOs. Known gaps at the position idx = 30 and 51Test stat the interpmissing method works with a mix of dising AOs. Known gaps at the position idx = 30 and 51Test that the interpmissing method works with a mix of dising AOs. Known gaps at the position idx = 30 and 51Test stat the interpmissing method works with a mix of dising AOs. Known gaps at the position idx = 30 and 51Test that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.06Tests that the interpmissing method properly applies hi			number in the input. 2) Check	
03Tests that the interpmissing method works with a matrix of AOs as input.Test that the interpmissing method works for a matrix of AOs as input. Known gaps at the position idx = 30 and 51pass method works for a matrix of AOs as input. Known gaps at the position idx = 30 and 5104Tests that the interpmissing method works with a list of AOs as input.Test that the interpmissing method works with a list of AOs as input.Test that the interpmissing method works with a list of AOs as input. Nown gaps at the position idx = 30 and 51pass04Tests that the interpmissing method works with a list of AOs as input.Test that the interpmissing method works for a list of AOs as input. Known gaps at the position idx = 30 and 51pass05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the result of applying to matrices and sectors and single AOs. Known gaps at the position idx = 30 and 51Test position idx = 30 and 5106Tests that the interpmissing method properly applies history.Test that the result of applying the interpmissing method can be processed back to an m-file.			that each output AO contains	
03 Tests that the interpmissing method works with a matrix of AOs as input. Test that the interpmissing method works for a matrix of AOs as input. Known gaps at the position idx = 30 and 51 pass 04 Tests that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works for a list of AOs as input. Known gaps at the position idx = 30 and 51 pass 04 Tests that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works with a list of AOs as input. Known gaps at the position idx = 30 and 51 pass 05 Tests that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method cond by position idx = 30 and 51 pass 06 Tests that the interpmissing method cond poperly applies history. Test that the result of applying pass pass 06 Tests that the interpmissing method cond by processed back to an m-file. Test that the result of applying pass pass			the correct data.	
03 method works with a matrix of AOs as input. method works for a matrix of AOs as input. Known gaps at the position idx = 30 and 51 1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data. pass 04 Tests that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works with a list of AOs as input. Test that the interpmissing method works for a list of AOs as input. Known gaps at the position idx = 30 and 51 pass 05 Tests that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the interpmissing method works with a mix of different shaped AOs as input. Test that the number of el- ments in 'out' is the same as in input. 2) Check that each output AO contains the correct data. pass 06 Tests that the interpmissing method properly applies history. Test that the result of applying pass pass	0.2	Tests that the interpmissing	Test that the interpmissing	pass
AOs as input.AOs as input. Known gaps at the position idx = 30 and 51pass1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.pass04Tests that the interpmissing method works with a list of AOs as input.Test that the interpmissing method works for a list of AOs as input. Known gaps at the position idx = 30 and 51pass05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with an input of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass06Tests that the interpmissing method properly applies history.Test that the result of applying the interpmissing method can be processed back to an m-file.pass	03	method works with a matrix of	method works for a matrix of	_
04Tests that the interprissing method works with a list of AOs as input.Test that the interprissing method works for a list of AOs as input.Test that the interprissing method works for a list of AOs as input.Test that the interprissing method works for a list of AOs as input.Test that the interprissing method works for a list of AOs as input.Test that the interprissing method works for a list of AOs as input.Test that the interprissing method works for a list of AOs as input.Test that the interprissing method works for a list of AOs as input.Test that the interprissing method works for a list of AOs as input.Test that the interprissing method works for a list of AOs as input.Test that the interprissing method works for a list of AOs as input.Test that the interprissing method works for a list of AOs as input.Test that the interprissing method works for a list of AOs as input.Test that the interprissing method works for a list of AOs as input.Test that the interprissing method works for a list of AOs as input.Test that the interprissing method works for a list of AOs as input.Test for AOS as input.<		AOs as input.	AOs as input. Known gaps at	
04Tests that the interpmissing method works with a list of AOs as input.1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.pass04Tests that the interpmissing method works with a list of AOs as input.Test that the interpmissing method works for a list of AOs as input. Known gaps at the position idx = 30 and 51pass05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass06Tests that the interpmissing method properly applies history.Test that the number of ele- ments in 'out' is the same as in input. 2) Check that each output AO contains the correct data.pass06Tests that the interpmissing method properly applies history.Test that the result of applying processed back to an m-file.pass		-	the position $idx = 30$ and 51	
04Tests that the interpmissing method works with a list of AOs as input.Test that the interpmissing method works with a list of AOs as input.Test that the interpmissing method works for a list of AOs as input. Known gaps at the position idx = 30 and 51pass05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of of method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of of method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of of method works with a mix of method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of of method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of of method works with a mix of of method works with a mix of method works with a mix of method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a minput of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass06Tests that the interpmissing method properly applies history.Test that the result of applying the interpmissing method can be processed back to an m-file.pass			1) Check that the number of ele-	pass
04Tests that the interpmissing method works with a list of AOs as input.Test that the interpmissing method works for a list of AOs as input.Test that the interpmissing method works for a list of AOs as input. Known gaps at the position idx = 30 and 51pass05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with an input of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with an input of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass06Tests that the interpmissing method properly applies history.Test that the result of applying the interpmissing method can be processed back to an m-file.pass			ments in 'out' is the square of the	_
04Tests that the interpmissing method works with a list of AOs as input.Test that he interpmissing method works for a list of AOs as input. Known gaps at the position idx = 30 and 51pass05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with an input of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with an input of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass06Tests that the interpmissing method properly applies history.Test that the interpmissing method can be processed back to an m-file.pass			number in the input. 2) Check	
04Tests that the interpmissing method works with a list of AOs as input.Test that the interpmissing method works for a list of AOs as input. Known gaps at the position idx = 30 and 51pass05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass06Tests that the interpmissing method properly applies history.Test that the result of applying the interpmissing method can be processed back to an m-file.pass			that each output AO contains	
04Tests that the interpmissing method works with a list of AOs as input.Test that the interpmissing method works for a list of AOs as input. Known gaps at the position idx = 30 and 51pass05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with an input of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass06Tests that the interpmissing method properly applies history.Test that the result of applying the interpmissing method can be processed back to an m-file.pass			the correct data.	
04method works with a list of AOs as input.method works for a list of AOs as input. Known gaps at the position idx = 30 and 511) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.pass05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with an input of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass06Tests that the interpmissing method properly applies history.Test that the interpmissing method can be processed back to an m-file.pass	0.4	Tests that the interpmissing	Test that the interpmissing	pass
as input.as input. Known gaps at the position idx = 30 and 51pass1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.pass05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with an input of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass06Tests that the interpmissing method properly applies history.Test that the result of applying the interpmissing method can be processed back to an m-file.pass	04	method works with a list of AOs	method works for a list of AOs	-
05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with an input of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass06Tests that the interpmissing method properly applies history.Test that the result of applying the interpmissing method can be processed back to an m-file.pass		as input.	as input. Known gaps at the	
05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with an input of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass06Tests that the interpmissing method properly applies history.Test that the result of applying the interpmissing method can be processed back to an m-file.pass		-	position $idx = 30$ and 51	
05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with an input of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass06Tests that the interpmissing method properly applies history.Test that the result of applying the interpmissing method can be processed back to an m-file.pass			1) Check that the number of ele-	pass
05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with an input of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass06Tests that the interpmissing method properly applies history.Test that the result of applying the interpmissing method can be processed back to an m-file.pass			ments in 'out' is the square of the	_
05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with an input of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass method single AOs.06Tests that the interpmissing method properly applies history.Test that the result of applying the interpmissing method can be processed back to an m-file.pass			number in the input. 2) Check	
05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with an input of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass06Tests that the interpmissing method properly applies history.Test that the result of applying processed back to an m-file.pass			that each output AO contains	
05Tests that the interpmissing method works with a mix of different shaped AOs as input.Test that the interpmissing method works with an input of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 51pass06Tests that the interpmissing method properly applies history.Test that the result of applying the interpmissing method can be processed back to an m-file.pass			the correct data.	
05method works with a mix of different shaped AOs as input.method works with an input of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 511) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.pass ements in 'out' is the same as in input. 2) Check that the result of applying the interpmissing method can be processed back to an m-file.	05	Tests that the interpmissing	Test that the interpmissing	pass
different shaped AOs as input.of matrices and vectors and single AOs. Known gaps at the position idx = 30 and 511) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.pass06Tests that the interpmissing method properly applies history.Test that the result of applying processed back to an m-file.pass	05	method works with a mix of	method works with an input	
$\begin{array}{ c c c c c c } \hline & single AOs. Known gaps at the position idx = 30 and 51 \\\hline & 1) Check that the number of eleements in 'out' is the same as in input. 2) Check that each output AO contains the correct data. \\\hline & 06 \\\hline & Tests that the interpmissing method properly applies history. \\\hline & 06 \\\hline & Tests that the interpmissing method can be processed back to an m-file. \\\hline & Tests that the interpmissing method can be processed back to an m-file. \\\hline & Tests that the interpmissing method can be processed back to an m-file. \\\hline & Tests that the interpmissing method can be processed back to an m-file. \\\hline & Tests that the interpmissing method can be processed back to an m-file. \\\hline & Tests that the interpmissing method can be processed back to an m-file. \\\hline & Tests that the interpmissing method can be processed back to an m-file. \\\hline & Tests that the interpmissing method can be processed back to an m-file. \\\hline & Tests that the interpmissing method can be processed back to an m-file. \\\hline & Tests that the method properly applies history. \\\hline & Test that the$		different shaped AOs as input.	of matrices and vectors and	
$\begin{array}{ c c c c c c }\hline 06 & \hline & position idx = 30 and 51 \\\hline & 1) \ Check that the number of elements in 'out' is the same as in input. 2) \ Check that each output AO contains the correct data. \\\hline & 06 & \hline & Tests that the interpmissing method properly applies history. \\\hline & & position idx = 30 and 51 \\\hline & 1) \ Check that the number of elements in 'out' is the same as in input. 2) \ Check that each output AO contains the correct data. \\\hline & & position idx = 30 and 51 \\\hline & & 1) \ Check that the number of elements in 'out' is the same as in input. 2) \ Check that each output AO contains the correct data. \\\hline & & position idx = 30 and 51 \\\hline & & position idx = 30 and 51 \\\hline & & position idx = 30 and 51 \\\hline & & position idx = 30 and 51 \\\hline & & position idx = 30 and 51 \\\hline & & position idx = 30 and 51 \\\hline & & position idx = 30 and 51 \\\hline & & position idx = 30 and 51 \\\hline & & position idx = 30 and 51 \\\hline & position $			single AOs. Known gaps at the	
06 1) Check that the number of elements in 'out' is the same as in input. 2) Check that each output AO contains the correct data. pass 06 Tests that the interpmissing method properly applies history. Test that the result of applying pass pass 06 Tests that the interpmissing method can be processed back to an m-file. Test that the result of applying pass pass			position $idx = 30$ and 51	
06 ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data. 06 Tests that the interpmissing method properly applies history. rests that the interpmissing method can be processed back to an m-file.			1) Check that the number of el-	pass
06 input. 2) Check that each output AO contains the correct data. 06 Tests that the interpmissing method properly applies history. Test that the result of applying the interpmissing method can be processed back to an m-file.			ements in 'out' is the same as in	
AO contains the correct data. 06 Tests that the interpmissing method properly applies history. Test that the result of applying the interpmissing method can be processed back to an m-file. pass			input. 2) Check that each output	
06Tests that the interpmissing method properly applies history.Test that the result of applying the interpmissing method can be processed back to an m-file.pass			AO contains the correct data.	
method properly applies history. the interpmissing method can be processed back to an m-file.	06	Tests that the interpmissing	Test that the result of applying	pass
processed back to an m-file.		method properly applies history.	the interpmissing method can be	
		· · · · · · · · · · · · · · · · · · ·	processed back to an m-file.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/interpmissing			
		1) Check that the last entry in the history of 'out' corresponds to 'interpmissing'. 2) Check that the re-built object is the same as 'out'.	pass
07	Tests that the interpmissing method can modify the input AO.	Test that the interpmissing method can modify the input AO by calling with no output and that the method doesn't change the input of the function notation (with a equal sign). Known gaps at the position idx = 30 and 51 1) Check that 'at1' and 'ain' are	pass
		is interprising(at1)	
08	Control the method with a plist.	Test that the interpmissing method keeps the data shape of the input object. The input AO must be an AO with row data and an AO with column data.	pass
		1) Check that the shpe of the	pass
09	Check that the interpmissing method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
10	Tests that the interpmissing method can change the tolerance for finding missing samples.	Test that the interpmissing method works with a plist which changes the tolerance. Known gaps at the position $idx = 30$ and 51 with the width of 40 and 60	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data. 3) Check that the re-built object is the same as 'out'.	pass
11	Check that the [ao/interpmissing] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/interpmissing		
	1) Check that the output con- tains the same plotinfo plist	pass

Table 61: Unit tests for ao/interpmissing.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



_ao/inv			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists	pass
	Tests that the inv method works	Test that the inv method works	naga
02	with a vector of AOs as input.	for a vector of AOs as input.	pass
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'atvec' 2) Check that each output	
		AO contains the correct data.	
03	Tests that the inv method works	Test that the inv method works	pass
	with a matrix of AOs as input.	for a matrix of AOs as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as	
		in 'atmat' 2) Check that each	
		output AO contains the correct	
		data.	
04	Tests that the inv method works	Test that the inv method works	pass
01	with a list of AOs as input.	for a list of AOs as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
05	Tests that the inv method works	Test that the inv method works	pass
	with a mix of different shaped	with an input of matrices and	
	AOs as input.	vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
06	Tests that the inv method	Test that the result of applying	pass
	property applies history.	the inv method can be processed	
		Dack to an m-nie.	
		the history of 'out' corresponde	pass
		the instory of out corresponds	
		built abject is the same as 'out'	
	Trata that the income the dama	The state of the s	
07	rests that the input AQ	rest that the input AO by call	pass
	modify the input AO.	ing with no output and that the	
		mg with no output and that the	
		of the function notation (with a	
		or the function notation (with a	
		1) Check that 'st4' and 'sin' and	negg
		now different 2) Check that 'sin'	pass
		$\operatorname{how}(\operatorname{unletent}, 2) \operatorname{Oleck}(\operatorname{unlet} all)$	
		15 111V (at 4).	1



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/inv			
08	Control the method with a plist.	Test that the inv method can modify the single axis controlled by the plist and the resuld can be	pass
		processed back to an m-file.	
		1) Check that the inv method	pass
		applies to the x-axis 2) Check	
		that the inv method applies to	
		the y-axis 3) Check that the inv	
		method applies to both axes 4)	
		Check that the re-built objects	
		are the same as 'out13'.	
00		Test that the inv method keeps	pass
09	Control the method with a plist.	the data shape of the input ob-	
		ject. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
10	Check that the inv method pass	Call the method with a list of	pass
10	back the output objects to a list	output variables and with a sin-	
	of output variables or to a single	gle output variable. Additionaly	
	variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/inv] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 62: Unit tests for ao/inv.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/isprop			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the isprop method works with a vector of AOs as input.	Test that the isprop method works for a vector of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atvec' 2) Check that each output contains the correct data.	pass
03	Tests that the isprop method works with a matrix of AOs as input.	Test that the isprop method works for a matrix of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atmat' 2) Check that each out- put contains the correct data.	pass
04	Tests that the isprop method works with a list of AOs as	Test that the isprop method works for a list of AOs as input.	pass
	input.	1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the isprop method works with a mix of different shaped AOs as input.	Test that the isprop method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the isprop method properly applies history.	The method isprop doesn't change the object, thus it is not necessary to apply history.	pass
	Trata that the important ad	Test that the issues mathed	pass
07	works for each property.	rest that the isprop method works for the properties: 'data', 'mfile', 'mfilename', 'mdlfile', 'mdlfilename', 'procinfo', 'plot- info', 'description', 'hist', 'name' 1) Check that each output con-	pass
		tains the correct data.	
08	Test the negatice case and the not function command.	Test that the isprop method retrun false for a unknown prop- erty and for methods of the ob- ject.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/isprop		
	1) Check that each output con-	pass
	tains the correct data.	

Table 63: Unit tests for ao/isprop.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



join/fsdata			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the join method	Test that the join method works	pass
02	works with a vector of AOs as	for a vector of AOs as input.	
	input.	1) Check that the output is ex-	pass
		act one AO. 2) Check that the	
		output have the correct data. 3)	
		Check the re-built object	
0.0	Tests that the join method	Tests that the join method works	pass
03	works with a matrix of AOs as	with a matrix of AOs as input.	
	input.	1) Check that the output is ex-	pass
		act one AO. 2) Check that the	
		output have the correct data. 3)	
		Check the re-built object	
	Tests that the join method	Tests that the join method works	pass
04	works with a list of AOs as	with a list of AOs as input.	1
	input.	1) Check that the output is ex-	pass
		act one AO. 2) Check that the	1
		output have the correct data. 3)	
		Check the re-built object	
0 -	Tests that the join method	Tests that the join method works	pass
05	works with a mix of different	with a mix of different shaped	-
	shaped AOs as input.	AOs as input.	
		1) Check that the output is ex-	pass
		act one AO. 2) Check that the	
		output have the correct data. 3)	
		Check the re-built object	
0.0	Tests that the join method	Test that the result of applying	pass
06	properly applies history.	the join method can be processed	1
		back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'join'. 2) Check that the re-	
		built object is the same object as	
		the input.	
07	Tests that the join method can	Test that the join method can	pass
07	modify the input AO.	modify the input AO by call-	1
		ing with no output and that the	
		method doesn't change the input	
		of the function notation (with a	
		equal sign).	
	1	<u> </u>	1



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

join/fsdata			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is joined 4) Check	
		that out and amodi are the same	
00	Test the shape of the output	Test that the join method keeps	pass
00	Test the shape of the output.	the data shape of the input ob-	
		ject. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shape of the	pass
		data doesn't change.	
11	Check that the [join/fsdata]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 64: Unit tests for join/fsdata.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



join/tsdata			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the join method	Test that the join method works	pass
02	works with a vector of AOs as	for a vector of AOs as input.	
	input.	1) Check that the output is ex-	pass
		act one AO. 2) Check that the	
		output have the correct data. 3)	
		Check the re-built object	
0.0	Tests that the join method	Test that the join method works	pass
03	works with a matrix of AOs as	for a matrix of AOs as input.	
	input.	1) Check that the output is ex-	pass
		act one AO. 2) Check that the	-
		output have the correct data. 3)	
		Check the re-built object	
	Tests that the join method	Tests that the join method works	pass
04	works with a list of AOs as	with a list of AOs as input.	1
	input.	1) Check that the output is ex-	pass
	I	act one AO. 2) Check that the	I
		output have the correct data. 3)	
		Check the re-built object	
	Tests that the join method	Tests that the join method works	pass
05	works with a mix of different	with a mix of different shaped	I
	shaped AOs as input.	AOs as input.	
	r r r r r r r r r r r r r r r r r r r	1) Check that the output is ex-	pass
		act one AO. 2) Check that the	P
		output have the correct data, 3)	
		Check the re-built object	
	Tests that the join method	Test that the result of applying	pass
06	properly applies history.	the join method can be processed	P
	FFJ	back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	T
		to 'join'. 2) Check that the re-	
		built object is the same object as	
		the input.	
	Tests that the join method can	Test that the join method can	pass
07	modify the input AO.	modify the input AO by call-	P COD
		ing with no output and that the	
		method doesn't change the input	
		of the function notation (with a	
		equal sign).	
I			



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

join/tsdata			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is joined 4) Check	
		that out and amodi are the same	
00	Test that the join method fills	Test that the join method fills	pass
08	the gaps with zeros.	the gaps with zeros.	
		1) Check that the join method	pass
		filled the gaps with zero 2) Check	
		that the re-built objects are the	
		same as out[12]	
00		Test that the join method keeps	pass
09	Test the shape of the output.	the data shape of the input ob-	-
		ject. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shape of the	pass
		data doesn't change.	
10	Tract the second base	Test the x-values.	pass
10	lest the x-values.	1) Check the x-values for the	pass
		case where we 'zerofill' or not.	
		2) Compare the outputs to see	
		that switching the order did not	
		matter, because if the different t0	
		of the data, that get sorted any-	
		ways.	
11	Check that the [join/tsdata]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
	_	works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 65: Unit tests for join/tsdata.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/lcohere			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the loohere method works with a vector of AOs as input.	Test that the loohere method works for a vector of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
03	Tests that the loohere method doesn't work with a matrix of AOs as input.	Test that the loohere method doesn't work for a matrix of AOs as input. 1) Nothing to do	pass
04	Tests that the loohere method works with a list of AOs as	Test that the loohere method works for a list of AOs as input.	pass
	input.	1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
05	Tests that the loohere method doesn't work with a mix of different shaped AOs as input.	Test that the loohere method doesn't work with an input of matrices and vectors and single AOs.	pass
06	Tests that the loohere method properly applies history.	Test that the result of applying the loohere method can be pro- cessed back to an m-file.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'lcohere'. 2) Check that the re- built object is the same object as 'out'.	pass
07	Tests that the loohere method can not modify the input AO.	Test that the loohere method can not modify the input AO. The method must throw an error for the modifier call.	pass
08	Test the shape of the output.	Test that the plus method keeps the data shape of the input ob- ject. The input AO must be an AO with row data and an AO with column data.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/lcohere			
		1) Check that the shpe of the	pass
		output data doesn't change.	
09	Check that the loohere method	This test is not longer necessary	pass
00	pass back the output objects to	because the cohere method pass	
	a list of output variables or to a	back always only one object.	
	single variable.	1) Nothing to check.	pass
11	Check that the [ao/lcohere]	Call the method with a list of	pass
11	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	-
10	Tests symmetry properties of	1) Prepare the test tsdata: white	pass
12	complex-coherence: 1) white	noise from normal distribution +	-
	noise produced from normal	offset 2) Assign a random unit	
	pdf, with a given mean value	3) Prepare the test tsdata: white	
	and sigma (distribution's 1st	noise from normal distribution +	
	and 2nd orders) 2) white noise	offset 4) Assign a random unit	
	produced from normal pdf, with	5) complex log-scale coherence of	
	a given mean value and sigma	the white noise	
	(distribution's 1st and 2nd	1) Check that $C(x,y)$ equals	pass
	orders) 3) complex looherence of	$\operatorname{coni}(C(\mathbf{v},\mathbf{x})) = 2$ Check that	T
	the white noise series 4)	C(x,x) equals 1 2) Check that	
	compare $C(x,y)$ with	C(y,y) equals 1	
10	TostsG(vnxhe5)vcproparei(S(afx)	1) Prepare the test tsdata: white	pass
13	earch fex, convertence: 1) white	noise from normal distribution +	-
	noise produced from normal	offset 2) Assign a random unit	
	pdf, with a given mean value	3) Prepare the test tsdata: white	
	and sigma (distribution's 1st	noise from normal distribution +	
	and 2nd orders) 2) white noise	offset 4) Assign a random unit 5)	
	produced from normal pdf, with	magnitude-squared log-scale co-	
	a given mean value and sigma	herence of the white noise	
	(distribution's 1st and 2nd	1) Check that $C(x,y)$ equals	pass
	orders) 3) magnitude-squared	C(y,x) 1) Check that $C(x,x)$	
	log-scale coherence of the white	equals 1 1) Check that $C(y,y)$	
	noise series 4) compare $C(x,y)$	equals 1	
14	Wetsbs G(mmhet) comparet iG(xok)	1) Prepare the test tsdata: white	pass
14	eouch for (x, y) white	noise from normal distribution +	
	noise produced from normal	offset 2) Assign a random unit	
	pdf, with a given mean value	3) Prepare the test tsdata: white	
	and sigma (distribution's 1st	noise from normal distribution +	
	and 2nd orders) 2) white noise	offset 4) Assign a random unit	
	produced from normal pdf, with	5) complex log-scale coherence of	
	a given mean value and sigma	the combination of noise	
	(distribution's 1st and 2nd	1) Check that the complex coher-	pass
	orders) 3) complex log-scale	ence equals 1	
	coherence of the combination of	· ·	1

coherence of the combination of white noise series 4) compare C(x,y) with 1



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/lcohere			
15	Tests symmetry properties of complex-coherence: 1) white noise produced from normal pdf, with a given mean value and sigma (distribution's 1st and 2nd orders) 2) white noise produced from normal pdf, with a given mean value and sigma (distribution's 1st and 2nd	 Prepare the test tsdata: white noise from normal distribution + offset 2) Assign a random unit Prepare the test tsdata: white noise from normal distribution + offset 4) Assign a random unit 5) magnitude-squared log-scale co- herence of the combination of noise 	pass
	orders) 3) magnitude-squared log-scale coherence of the	1) Check that the magnitude- squared coherence equals 1	pass
16	Contribution try typeroise of corrigslex - cohrpon ceC (k), w) hitith 1 noise produced from normal pdf, with a given mean value and sigma (distribution's 1st and 2nd orders) 2) white noise produced from normal pdf, with a given mean value and sigma (distribution's 1st and 2nd orders) 3) magnitude-squared	 Prepare the test tsdata: white noise from normal distribution + offset 2) Assign a random unit Prepare the test tsdata: white noise from normal distribution + offset 4) Assign a random unit 5) magnitude-squared log-scale co- herence of the combination of noise 6) complex log-scale coher- ence of the combination of noise 	pass
	log-scale coherence M of the combination of white noise series 4) complex log-scale coherence C of the combination	1) Check that the magnitude- squared coherence equals the square modulus of the complex coherence	pass
17	Westshamdingsofies : (c)) white abis(C)) 2 oviited Mfrom normal pdf, with a given mean value and sigma (distribution's 1st and 2nd orders) 2) white noise produced from normal pdf, with a given mean value and sigma	 Prepare the test tsdata: white noise from normal distribution + offset 2) Assign a random unit Prepare the test tsdata: white noise from normal distribution + offset 4) Assign a random unit 5) complex cohere of the white noise 	pass
	(distribution's 1st and 2nd orders) 3) complex cohere of the white noise series 4) compares the units of the input and	1) Check that (complex coher- ence yunits) equals [1] 2) Check that (complex coherence xunits) equals [Hz]	pass
18	Gespathandling of units: 1) white noise produced from normal pdf, with a given mean value and sigma (distribution's 1st and 2nd orders) 2) white noise produced from normal pdf, with a given mean value and sigma (distribution's 1st and 2nd	 Prepare the test tsdata: white noise from normal distribution + offset 2) Assign a random unit Prepare the test tsdata: white noise from normal distribution + offset 4) Assign a random unit 5) magnitude-squared cohere of the white noise 	pass
	orders) 3) magnitude-squared cohere of the white noise series 4) compares the units of the input and output	 Check that (magnitude-squared coherence yunits) equals 2) Check that (magnitude-squared coherence xunits) equals [Hz] 	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/lcohere			
30	Tests handling of special cases:	1) Prepare the test tsdata: white	pass
30	1) white noise produced from	noise from normal distribution +	
	normal pdf, with a given mean	offset 2) Assign a random unit 3)	
	value and sigma (distribution's	Prepare the test tsdata: the same	
	1st and 2nd orders) 2) the same	data as 1) and 2) 4) loohere of the	
	noise series 3) loohere of the	series	
	white noise series 4) compares	1) Check that calculated loohere	pass
	the output to unity	equals 1	

Table 66: Unit tests for ao/lcohere.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/lcpsd			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the lcpsd method works with a vector of AOs as input.	Test that the lcpsd method works for a vector of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
03	Tests that the lcpsd method doesn't work with a matrix of AOs as input.	Test that the lcpsd method doesn't work for a matrix of AOs as input. 1) Nothing to check.	pass
04	Tests that the lcpsd method works with a list of AOs as	Test that the lcpsd method works for a list of AOs as input.	pass
	input.	1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
05	Tests that the lcpsd method doesn't work with a mix of different shaped AOs as input.	Test that the lcpsd method doesn't work with an input of matrices and vectors and single AOs. 1) Nothing to check.	pass
06	Tests that the lcpsd method properly applies history.	Test that the result of applying the lcpsd method can be pro- cessed back to an m-file.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'lcpsd'. 2) Check that the re- built object is the same object as 'out'.	pass
07	Tests that the lcpsd method can not modify the input AO.	Test that the lcpsd method can not modify the input AO. The method must throw an error for the modifier call.	pass
08	Test the shape of the output.	Test that the plus method keeps the data shape of the input ob- ject. The input AO must be an AO with row data and an AO with column data.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/lcpsd			
		1) Check that the shpe of the	pass
		output data doesn't change.	
00	Check that the lcpsd method	This test is not longer necessary	pass
03	pass back the output objects to	because the cohere method pass	
	a list of output variables or to a	back always only one object.	
	single variable.	1) Nothing to check	pass
11	Check that the [ao/lcpsd]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	
19	Tests that differently sized data	Test that applying lcpsd works	pass
12	sets are treated properly	on two AOs.	
		1) Check that lcpsd used the	pass
		length of the shortest ao.	
17	Tests handling of units: 1) white	1) Prepare the test tsdata: white	pass
	noise produced from normal	noise from normal distribution +	
	pdf, with a given mean value	offset 2) Assign a random unit	
	and sigma (distribution's 1st	3) Prepare the test tsdata: white	
	and 2nd orders) 2) white noise	noise from normal distribution +	
	produced from normal pdf, with	offset 4) Assign a random unit 5)	
	a given mean value and sigma	LCPSD of the white noise	
	(distribution's 1st and 2nd	1) Check that (calculated	pass
	orders) 3) LCPSD of the white	LCPSD yunits) equals input_1	
	noise series 4) compares the	$units*input_2 units/Hz$	
18	Tests datablingport amidsout)produite	1) Prepare the test tsdata: white	pass
10	noise produced from normal	noise from normal distribution +	
	pdf, with a given mean value	offset 2) Assign a random unit	
	and sigma (distribution's 1st	3) Prepare the test tsdata: white	
	and 2nd orders) 2) white noise	noise from normal distribution +	
	produced from normal pdf, with	offset 4) Assign a random unit	
	a given mean value and sigma	5) LCPSD of the white noise 6)	
	(distribution's 1st and 2nd	LPSD of the white noise	
	orders) 3) LCPSD of the white	1) Check that $LCPSD(x,x)$	pass
	noise series Comparison with	equals $LPSD(x) 2$ Check that	
	LPSD: 4) compares the	LCPSD(y,y) equals $LPSD(y)$ 3)	
	off-diagonal terms to check they	Check that $LCPSD(x,y)$ equals	
	are complex-conjugated 5)	conj(LCPSD(y,x))	

compares the diagonal terms with PSD of the individual noise Table 67: Unit tests for ao/lcpsd.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/le			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.0	Tests that the le method	Test that the le method works	pass
02	compare an AO with scalar	with relational operators and the	
	value	function command. Use for this	
		AOs with different data objects.	
		1) Check that the result of	pass
		the 'relational operator' and	
		the 'function' command are the	
		same. 2) Check that each output	
		AO contains the correct data.	
02	Tests that the le method	Test that the le method works	pass
05	compare an AO with one other	with relational operators and the	
	AO	function command. Use for this	
		AOs with different data objects.	
		Remark that both AOs must	
		have the same size.	
		1) Check that the result of	pass
		the 'relational operator' and	
		the 'function' command are the	
		same. 2) Check that each output	
		AO contains the correct data.	

Table 68: Unit tests for ao/le.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/len			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the len method works	Test that the len method works	pass
02	with a vector of AOs as input.	for a vector of AOs as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'atvec' 2) Check that each output	
		contains the correct data.	
03	Tests that the len method works	Test that the len method works	pass
	with a matrix of AOs as input.	for a matrix of AOs as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as	
		in 'atmat' 2) Check that each	
		output AO contains the correct	
		data.	
04	Tests that the len method works	Test that the len method works	pass
04	with a list of AOs as input.	for a list of AOs as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
05	Tests that the len method works	Test that the len method works	pass
00	with a mix of different shaped	with an input of matrices and	
	AOs as input.	vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
06	Tests that the len method	The len method doesn't change	pass
	properly applies history.	the object, thus it is not neces-	
		sary to test the history. Nothing	
		to do.	
			pass

Table 69: Unit tests for a o/len.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/linSubtract			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the linSubtract method works with a vector of AOs as input.	Test that the linSubtract method works for a vector of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
03	Tests that the linSubtract method works with a matrix of AOs as input.	Test that the linSubtract method works for a matrix of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
04	Tests that the linSubtract method works with a list of AOs	Test that the linSubtract method works for a list of AOs as input.	pass
	as input.	1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
05	Tests that the linSubtract method works with a mix of different shaped AOs as input.	Test that the linSubtract method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the linSubtract method properly applies history.	Test that the result of applying the linSubtract method can be processed back to an m-file.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'linSubtract'. 2) Check that the re-built object is the same object as 'out'.	pass
07	Tests that the linSubtract method can not modify the input AO.	Test that the tfe method can not modify the input AO. The method must throw an error for the modifier call.	pass
		1) Nothing to check.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/linSubtract			
11	Check that the [ao/linSubtract]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 70: Unit tests for ao/linSubtract.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/lincom			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the lincom method works with a vector of AOs as input.	Test that the lincom method works for a vector of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is 1 2) Check that the output AO contains the correct units 3) Check that the output AO contains the correct data.	pass
04	Tests that the lincom method works with a list of AOs as input.	Test that the lincom method works for a list of AOs as input. 1) Check that the number of el- ements in 'out' is 1. 2) Check that the output AO contains the correct units 3) Check that the output AO contains the correct data.	pass
06	Tests that the lincom method properly applies history.	Test that the result of applying the lincom method can be pro- cessed back to an m-file. 1) Check that the last entry in the history of 'out' corresponds to 'lincom'. 2) Check that the re- built object is the same object as	pass
07	Tests that the lincom method can not modify the input AO.	'out'. Test that the lincom method can not modify the input AO. The method must throw an error for the modifier call.	pass
08	Test the shape of the output.	 Nothing to check. Test that the plus method keeps the data shape of the input object. The input AO must be an AO with row data and an AO with column data. Check that the shpe of the output data doesn't change. 	pass pass pass
11	Check that the [ao/lincom] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/lincom			
		1) Check that the output con-	pass
		tains the same plotinfo plist	
12	Tests that the lincom method works with a vector of AOs as input $+$ a pest object.	Test that the lincom method works for a vector of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is 1 2) Check that the output AO contains the correct units 3) Check that the output AO contains the correct data	pass
14	Tests that the lincom method works with a list of AOs as	Test that the lincom method works for a list of AOs as input.	pass
	input + a pest object.	1) Check that the number of el- ements in 'out' is 1. 2) Check that the output AO contains the correct units 3) Check that the output AO contains the correct data.	pass

Table 71: Unit tests for a o/lincom.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/ln			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the [ao/ln] method	Test that the [ao/ln] method	pass
02	works with a vector of objects	works for a vector of objects as	
	as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'vec' 2) Check that each output	
		object contains the correct data.	
03	Tests that the [ao/ln] method	Test that the [ao/ln] method	pass
00	works with a matrix of objects	works for a matrix of objects as	
	as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
04	Tests that the [ao/ln] method	Test that the [ao/ln] method	pass
04	works with a list of objects as	works for a list of objects as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/ln] method	Tests that the [ao/ln] method	pass
00	works with a mix of different	works with a mix of different ar-	
	arrays of objects as input.	rays of objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
06	Tests that the [ao/ln] method	Test that the result of applying	pass
	properly applies history.	the [ao/ln] method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to ' $[ao/ln]$ '. 2) Check that the re-	
		built object is the same object as	
		the input.	
07	Tests that the [ao/ln] method	Test that the [ao/ln] method can	pass
07	can modify the input AO.	modify the input object by call-	-
	v 1	ing with no output and that the	
		method doesn't change the input	
		of the function notation (with a	
		equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/ln			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that	
		the modified input is the [ao/ln]	
		value of the copy 4) Check that	
		out and amodi are the same	
00	Test the shape of the data in	Test that the [ao/ln] method	pass
09	AOs.	keeps the data shape of the in-	
		put object. The input AO data	
		must be an array with row data	
		and/or column data.	
		1) Check that the shape of the	pass
		data doesn't change.	
10	Check that the [ao/ln] method	Call the method with a list of	pass
10	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/ln] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 72: Unit tests for a o/ln.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/loadobj			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check the shape of the loaded	pass
		objects.	

Table 73: Unit tests for ao/loadobj.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/log			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the [ao/log] method	Test that the [ao/log] method	pass
02	works with a vector of objects	works for a vector of objects as	
	as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'vec' 2) Check that each output	
		object contains the correct data.	
03	Tests that the [ao/log] method	Test that the [ao/log] method	pass
	works with a matrix of objects	works for a matrix of objects as	
	as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
04	Tests that the [ao/log] method	Test that the [ao/log] method	pass
01	works with a list of objects as	works for a list of objects as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		(mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/log] method	Tests that the [ao/log] method	pass
	works with a mix of different	works with a mix of different ar-	
	arrays of objects as input.	rays of objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		mat ²) Cneck that each output	
		object contains the correct data.	
06	near and a series is the former and the series of the seri	the lee (lee) method applying	pass
	property applies history.	the [ao/log] method can be pro-	
		1) Check that the last entry in	
		1) Check that the last entry in the history of 'out' companyed	pass
		the mistory of out corresponds	
		ro huilt object is the same object	
		as the input	
	Tosta that the [as /log] method	Tost that the loc/log method	negg
07	can modify the input AQ	an modify the input chiest be	pass
	can mouny the input AO.	calling with no output and that	
		the method decen't shapes the	
		input of the function notation	
		(with a coupl give)	
		(with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/log			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that	
		the modified input is the [ao/log]	
		value of the copy 4) Check that	
		out and amodi are the same	
00	Test the shape of the data in	Test that the [ao/log] method	pass
09	AOs.	keeps the data shape of the in-	-
		put object. The input AO data	
		must be an array with row data	
		and/or column data.	
		1) Check that the shape of the	pass
		data doesn't change.	1
10	Check that the [ao/log] method	Call the method with a list of	pass
10	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/log] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
	-	works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 74: Unit tests for ao/log.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/log10			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set	pass
		1) Charle that wether and a set of the set o	
		1) Check that getinio call re-	pass
		turned an minio object in an	
		cases. 2) Check that all plists	
	$T_{r} = t_{r} + b_{r} + b_{r} - [a_{r} - b_{r}]$	The the correct parameters.	
02	rests that the [a0/10g10]	rest that the [ao/log10] method	pass
	method works with a vector of	works for a vector of objects as	
	objects as input.	1) Chock that the number of al	nagg
		amonta in 'out' is the same as in	pass
		'was' 2) Check that each output	
		object contains the correct data	
	Tagta that the [so/log10]	Test that the [ac/log10] method	naga
03	method works with a matrix of	works for a matrix of objects as	pass
	chiests as input	input	
	objects as input.	1) Chock that the number of al	naga
		amonta in 'out' is the same as in	pass
		'mat' 2) Check that each output	
		object contains the correct data	
	Tosts that the [so/log10]	Test that the [ac/log10] method	naga
04	mothed works with a list of	works for a list of objects as in	pass
	objects as input	but	
	objects as input.	1) Check that the number of el-	nass
		ements in 'out' is the same as in	pass
		'mat' 2) Check that each output	
		object contains the correct data	
	Tests that the [ao/log10]	Tests that the $[ao/log10]$ method	nass
05	method works with a mix of	works with a mix of different ar-	pass
	different arrays of objects as	rays of objects as input	
	input	1) Check that the number of el-	nass
	input.	ements in 'out' is the same as in	pass
		'mat' 2) Check that each output	
		object contains the correct data.	
	Tests that the [ao/log10]	Test that the result of apply-	pass
06	method properly applies history.	ing the [ao/log10] method can be	Pass
		processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	T
		to $(ao/log10)$. 2) Check that the	
		re-built object is the same object	
		as the input.	
	Tests that the [ao/log10]	Test that the [ao/log10] method	pass
07	method can modify the input	can modify the input object by	1
	AO.	calling with no output and that	
		the method doesn't change the	
		input of the function notation	
		(with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/log10			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the [ao/log10]	
		value of the copy 4) Check that	
		out and amodi are the same	
00	Test the shape of the data in	Test that the [ao/log10] method	pass
09	AOs.	keeps the data shape of the input	-
		object. The input AO data must	
		be an array with row data and/or	
		column data.	
		1) Check that the shape of the	pass
		data doesn't change.	-
10	Check that the [ao/log10]	Call the method with a list of	pass
10	method pass back the output	output variables and with a sin-	-
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
	Ŭ	works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	-
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/log10]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 75: Unit tests for a o/log10.


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/lpsd			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the lpsd method works with a vector of AOs as input.	Test that the lpsd method works for a vector of AOs as input. 1) Check that the number of el- ements in 'out' is the same as in the input. 2) Check that each output object contains the cor- rect values.	pass
03	Tests that the lpsd method works with a matrix of AOs as input.	 Test that the lpsd method works for a matrix of AOs as input. 1) Check that the number of el- ements in 'out' is the same as in the input. 2) Check that each output object contains the cor- rect values. 	pass
04	Tests that the lpsd method works with a list of AOs as input.	 Test that the lpsd method works for a list of AOs as input. 1) Check that the number of el- ements in 'out' is the same as in the input. 2) Check that each output AO contains the correct data. 	pass
05	Tests that the lpsd method works with a mix of different shaped AOs as input.	 Test that the lpsd method works with an input of matrices and vectors and single AOs. 1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data. 	pass
06	Tests that the lpsd method properly applies history.	 Test that the result of applying the lpsd method can be processed back to an m-file. 1) Check that the last entry in the history of 'out' corresponds to 'lpsd'. 2) Check that the rebuilt object is the same object as 'out'. 	pass
07	Check that the lpsd method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/lpsd			
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
08	Tests that the lpsd method agrees with MATLAB's pwelch if the frequency resolution of the latter is changed to match those used in lpsd	Test that the applying psd works on a single AO. 1) Check that the DFT equa- tions (eq. (3) and (4) in [1]) are fulfilled. 2) Check that we get the same outputs for each frequency bin when computing the psd using matlab's fft with the frequency resolution values retrieved by lpsd (and already tested in (1)) [1] G. Heinzel, lpsd revisited: ltf, S2-AEI-TN-3052 Call the method with a list of	pass
11	Check that the [ao/198d] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a single output variable. Additionaly check that the rebuild method works on the output.1) Check that the output contains the same plotinfo plist	pass
17	Tests handling of units: 1) white noise produced from normal pdf, with a given mean value and sigma (distribution's 1st and 2nd orders) 2) LPSD of the white noise 3) compares the units of the input and output	 Prepare the test tsdata: white noise from normal distribution + offset 2) Assign a random unit 3) LPSD of the white noise Check that (calculated LPSD yunits) equals (input units)2 / Hz 2) Check that (calculated LPSD xunits) equals Hz 	pass
18	Tests handling of units: 1) white noise produced from uniform pdf, with a given mean value and sigma (distribution's 1st and 2nd orders) 2) LASD of the white noise 3) compares the units of the input and output	 Prepare the test tsdata: white noise from uniform distribution + offset 2) Assign a random unit LASD of the white noise Check that (calculated LASD yunits) equals (input units) / Hz(1/2) 2) Check that (calculated LASD xunits) equals Hz 	pass
19	Tests handling of units: 1) white noise produced from normal pdf, with a given mean value and sigma (distribution's 1st and 2nd orders) 2) LPS of the white noise 3) compares the units of the input and output	 Prepare the test tsdata: white noise from normal distribution + offset 2) Assign a random unit 3) LPS of the white noise Check that (calculated LPS yunits) equals (input units)2 2) Check that (calculated LPS xu- nits) equals Hz 	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/lpsd			
20	Tests handling of units: 1) white	1) Prepare the test tsdata: white	pass
20	noise produced from uniform	noise from uniform distribution	
	distribution, with a given mean	+ offset 2) Assign a random unit	
	value and sigma (distribution's	3) LAS of the white noise	
	1st and 2nd orders) 2) LAS of	1) Check that (calculated LAS	pass
	the white noise 3) compares the	yunits) equals (input units) 2)	
	units of the input and output	Check that (calculated LAS xu-	
		nits) equals Hz	
41	Tests the effect of windowing: 1)	1) Prepare the test tsdata: white	pass
	white noise produced from	noise from normal distribution +	
	normal pdf, with: a given mean	offset 2) Calculate the statistical	
	value and sigma (distribution's	parameters 3) Estimate the psd	
	1st and 2nd order) 2) lpsd	without detrending, Rectangular	
	passing the window name	window (name) 4) Estimate the	
	(Rectangular) 3) lpsd passing	psd without detrending, Rectan-	
	the window object (Rectangular	gular window (object) 5) Com-	
	type) 4) compares the two psds	pare results	
		1) Check that calculated psds are	pass
		identical	
42	Tests the effect of windowing: 1)	1) Prepare the test tsdata: white	pass
	white noise produced from	noise from normal distribution +	
	normal pdf, with: a given mean	offset 2) Calculate the statistical	
	value and sigma (distribution's	mithaut data data price price	
	Ist and 2nd order) 2) lpsd	dow (norma) 4) Estimate the rad	
	(DIIO2) 2) logd pagging the	dow (name) 4) Estimate the psd	
	(BH92) 5) Ipsd passing the	dow (abject) 5) Company populta	
	acmpares the two pads	1) Check that calculated pade are	page
	compares the two psus	identical	pass
	Tests the effect of windowing: 1)	1) Prepare the test tsdata: white	nass
43	white noise produced from	noise from normal distribution +	pass
	normal pdf, with: a given mean	offset 2) Calculate the statisti-	
	value and sigma (distribution's	cal parameters 3) Estimate the	
	1st and 2nd order) 2) lpsd	psd without detrending. Ham-	
	passing the window name	ming window (name) 4) Esti-	
	(Hamming) 3) lpsd passing the	mate the psd without detrend-	
	window object (Hamming type)	ing, Hamming window (object)	
	4) compares the two psds	5) Compare results	
		1) Check that calculated psds are	pass
		identical	
44	Tests the effect of windowing: 1)	1) Prepare the test tsdata: white	pass
44	white noise produced from	noise from normal distribution +	
	normal pdf, with: a given mean	offset 2) Calculate the statisti-	
	value and sigma (distribution's	cal parameters 3) Estimate the	
	1st and 2nd order) 2) lpsd	psd without detrending, Hanning	
	passing the window name	window (name) 4) Estimate the	
	(Hanning) 3) lpsd passing the	psd without detrending, Hanning	
	window object (Hanning type)	window (object) 5) Compare re-	
	4) compares the two psds	sults	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/lpsd			
		1) Check that calculated psds are identical	pass
45	Tests the effect of windowing: 1) white noise produced from normal pdf, with: a given mean value and sigma (distribution's 1st and 2nd order) 2) lpsd passing the window name (Bartlett) 3) lpsd passing the window object (Bartlett type) 4) compares the two psds	 Prepare the test tsdata: white noise from normal distribution + offset 2) Calculate the statisti- cal parameters 3) Estimate the psd without detrending, Bartlett window (name) 4) Estimate the psd without detrending, Bartlett window (object) 5) Compare re- sults Check that calculated psds are 	pass
		identical	Pass
46	Tests the effect of windowing: 1) white noise produced from normal pdf, with: a given mean value and sigma (distribution's 1st and 2nd order) 2) lpsd passing the window name (Nuttall3) 3) lpsd passing the window object (Nuttall3 type) 4) compares the two psds	1) Prepare the test tsdata: white noise from normal distribution + offset 2) Calculate the statisti- cal parameters 3) Estimate the psd without detrending, Nuttall3 window (name) 4) Estimate the psd without detrending, Nuttall3 window (object) 5) Compare re- sults	pass
		1) Check that calculated psds are identical	pass
47	Tests the effect of windowing: 1) white noise produced from normal pdf, with: a given mean value and sigma (distribution's 1st and 2nd order) 2) lpsd passing the window name (Kaiser, psll = 150) 3) lpsd passing the window object	1) Prepare the test tsdata: white noise from normal distribution + offset 2) Calculate the statistical parameters 3) Estimate the psd without detrending, Kaiser win- dow (name) 4) Estimate the psd without detrending, Kaiser win- dow (object) 5) Compare results	pass
	(Kaiser type, $psll = 150$) 4) compares the two $psds$	1) Check that calculated psds are identical	pass
48	Tests the effect of windowing: 1) white noise produced from normal pdf, with: a given mean value and sigma (distribution's 1st and 2nd order) 2) lpsd passing the window name (Kaiser, psll = default) 3) lpsd passing the window object	1) Prepare the test tsdata: white noise from normal distribution + offset 2) Calculate the statistical parameters 3) Estimate the psd without detrending, Kaiser win- dow (name) 4) Estimate the psd without detrending, Kaiser win- dow (object) 5) Compare results	pass
	(Kaiser type, psll = default) 4) compares the two psds	1) Check that calculated psds are identical	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/lpsd			
49	Tests the effect of windowing: 1)	1) Prepare the test tsdata: white	pass
45	white noise produced from	noise from normal distribution +	
	normal pdf, with: a given mean	offset 2) Calculate the statisti-	
	value and sigma (distribution's	cal parameters 3) Estimate the	
	1st and 2nd order) 2) lpsd	psd without detrending, Nuttall4	
	passing the window name	window (name) 4) Estimate the	
	(Nuttall4) 3) lpsd passing the	psd without detrending, Nuttall4	
	window object (Nuttall4 type)	window (object) 5) Compare re-	
	4) compares the two psds	sults	
		1) Check that calculated psds are	pass
		identical	
50	Tests the effect of windowing: 1)	1) Prepare the test tsdata: white	pass
00	white noise produced from	noise from normal distribution +	
	normal pdf, with: a given mean	offset 2) Calculate the statistical	
	value and sigma (distribution's	parameters 3) Estimate the psd	
	1st and 2nd order) 2) lpsd	without detrending, SFT3F win-	
	passing the window name	dow (name) 4) Estimate the psd	
	(SFT3F) 3) lpsd passing the	without detrending, SFT3F win-	
	window object (SFT3F type) 4)	dow (object) 5) Compare results	
	compares the two psds	1) Check that calculated psds are	pass
		identical	

Table 76: Unit tests for ao/lpsd.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/lscov			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the lscov method works with a vector of AOs as	Test that the lscov method works for a vector of AOs as input.	pass
	mput.	1) Check the data type of the output 2) Check that each out- put AO contains the correct data.	pass
03	Tests that the lscov method works with a matrix of AOs as input.	Tests that the lscov method works with a matrix of AOs as input.	pass
		1) Check the data type of the output 2) Check that each out- put AO contains the correct data.	pass
04	Tests that the lscov method works with a list of AOs as input.	Tests that the lscov method works with a list of AOs as in- put.	pass
		1) Check the data type of the output 2) Check that each out- put AO contains the correct data.	pass
05	Tests that the lscov method properly applies history.	Test that the result of applying the lscov method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'lscov'. 2) Check that the re- built object is the same object as the input.	pass
06	The lscov method can not be used as a modifer method.	The lscov method can not be used as a modifer method. 1) Nothing to check.	pass
07	Check that the lscov method uses weights for the fit.	Check that the lscov method uses weights for the fit.	pass
		1) Check the output data 2) Check the yunits 3) Check that 'out1' and 'out2' have the same data 4) Check that 'out3' and 'out4' have the same data 5) Check the re-built objects	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/lscov			
11	Check that the [ao/lscov]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 77: Unit tests for ao/lscov.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/lt			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
00	Tests that the lt method	Test that the lt method works	pass
02	compare an AO with scalar	with relational operators and the	-
	value	function command. Use for this	
		AOs with different data objects.	
		1) Check that the result of	pass
		the 'relational operator' and	
		the 'function' command are the	
		same. 2) Check that each output	
		AO contains the correct data.	
0.2	Tests that the lt method	Test that the lt method works	pass
03	compare an AO with one other	with relational operators and the	-
	AO	function command. Use for this	
		AOs with different data objects.	
		Remark that both AOs must	
		have the same size.	
		1) Check that the result of	pass
		the 'relational operator' and	
		the 'function' command are the	
		same. 2) Check that each output	
		AO contains the correct data.	

Table 78: Unit tests for ao/lt.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/ltfe			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all	pass
		cases 2) Check that all plists	
		have the correct parameters.	
	Tests that the ltfe method works	Test that the ltfe method works	pass
02	with a vector of AOs as input.	for a vector of AOs as input.	
		1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
03	Tests that the ltfe method	Test that the ltfe method doesn't	pass
	doesn't work with a matrix of	work for a matrix of AOs as in-	
	AOs as input.	put.	
		1) Nothing to check.	pass
04	Tests that the life method works	Test that the ltfe method works	pass
	with a list of AOs as input.	1) Check that the number of ele	magg
		1) Check that the number of ele-	pass
		number in the input 2) Check	
		that each output AO contains	
		that each output AO contains	
	Tests that the ltfe method	Test that the ltfe method doesn't	pass
05	doesn't work with a mix of	work with an input of matrices	Pabb
	different shaped AOs as input.	and vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
06	Tests that the ltfe method	Test that the result of applying	pass
	properly applies history.	the ltfe method can be processed	
		back to an m-file.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'ltfe'. 2) Check that the re-	
		built object is the same object as	
		The state of the s	
07	not modify the input AO	not modify the input AO. The	pass
	not mouny the input AO.	mothod must throw an error for	
		the modifier call	
		1) Nothing to check	pass
		Test that the ltfe method keeps	pass
08	Test the shape of the output.	the data shape of the input ob-	Pass
		iect. The input AO must be an	
		AO with row data and an AO	
		with column data.	
1	I		



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/ltfe			
		1) Check that the shpe of the	pass
		output data doesn't change.	
00	Check that the ltfe method pass	This test is not longer necessary	pass
09	back the output objects to a list	because the ltfe method pass	
	of output variables or to a single	back always only one object.	
	variable.	1) Nothing to check	pass
11	Check that the [ao/ltfe] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	
17	Tests handling of units: 1) white	1) Prepare the test tsdata: white	pass
11	noise produced from normal	noise from normal distribution +	
	pdf, with a given mean value	offset 2) Assign a random unit	
	and sigma (distribution's 1st	3) Prepare the test tsdata: white	
	and 2nd orders) 2) white noise	noise from normal distribution +	
	produced from normal pdf, with	offset 4) Assign a random unit 5)	
	a given mean value and sigma	Ltfe of the white noise	
	(distribution's 1st and 2nd	1) Check that (calculated ltfe yu-	pass
	orders) 3) Ltfe of the white	nits) equals [1/Hz]	
20	Teiste serieslishg comspecial teases:	1) Prepare the test tsdata: white	pass
50	1) in the the is produced of the the second se	noise from normal distribution +	
	normal pdf, with a given mean	offset 2) Assign a random unit 3)	
	value and sigma (distribution's	Prepare the test tsdata: the same	
	1st and 2nd orders) 2) the same	data as 1) and 2) 4) ltfe of the	
	noise series 3) ltfe of the white	series	
	noise series 4) compares the	1) Check that calculated ltfe	pass
	output to unity	equals 1	

Table 79: Unit tests for a o/ltfe.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/max			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the [ao/max] method	Test that the [ao/max] method	pass
02	works with a vector of objects	works for a vector of objects as	
	as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'vec' 2) Check that each output	
		object contains the correct data.	
03	Tests that the [ao/max] method	Test that the [ao/max] method	pass
	works with a matrix of objects	works for a matrix of objects as	
	as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
04	Tests that the [ao/max] method	Test that the [ao/max] method	pass
	works with a list of objects as	works for a list of objects as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/max] method	Tests that the [ao/max] method	pass
	works with a mix of different	works with a mix of different ar-	
	arrays of objects as input.	rays of objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		mat ²) Check that each output	
	Tests that the [as (max] mothed	The that the result of applying	magg
06	rests that the [ao/max] method	the lee (meril method can be pro-	pass
	property applies instory.	accord back	
		1) Check that the last entry in	nagg
		the history of 'out' corresponde	pass
		the instory of out corresponds	
		ro built object is the same object	
		as the input	
	Tosts that the [so/max] method	Tost that the [ac/max] method	negg
07	$\begin{bmatrix} 100 \\ 100 \end{bmatrix}$ can modify the input $\Delta \Omega$	can modify the input object by	pass
		calling with no output and that	
		the method doesn't change the	
		input of the function notation	
		(with a equal sign)	
		(with a equal sign).	1



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/max			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the [ao/max]	
		value of the copy 4) Check that	
		out and amodi are the same	
200	Test that the [ao/max] method	Test that the [ao/max] method	pass
208	uses the plist to get the axis.	uses the plist to get the axis.	-
	This is intended to test methods	1) Check that the [ao/max]	pass
	like ao/max and ao/min which	method applies to the x-axis 2)	1
	only allow 'x' and 'y' choices.	Check that the [ao/max] method	
		applies to the v-axis 3) Check	
		that the [ao/max] method ap-	
		plies to both axes 4) Check that	
		the re-built object is the same as	
		in 'out[13]'.	
00	Test the shape of the data in	Test that the [ao/max] method	pass
09	AOs.	keeps the data shape of the input	-
		object. The input AO data must	
		be an array with row data and/or	
		column data.	
		1) Check that the shape of the	pass
		data doesn't change.	-
10	Check that the [ao/max]	Call the method with a list of	pass
10	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/max]	Call the method with a list of	pass
11	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 80: Unit tests for a o/max.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mcmc			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
-	for this method.	for no sets, all sets, and each set	
		1) Check that get Info call re	20000
		1) Check that getimo can re-	pass
		appendix an infino object in an	
		have the correct parameters	
	Tests that the metropolis2D	Test that the metropolis2D	pass
02	method works with a vector of	method works for a vector of	pass
	AOs as input.	AOs as input.	
	T, T, T, T,	1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
03	Tests that the metropolis2D	Test that the metropolis2D	pass
05	method works with a matrix of	method works for a matrix of	
	AOs as input.	AOs as input.	
		1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
04	Tests that the metropolis2D	Test that the metropolis2D	pass
	method works with a list of AOs	method works for a list of AOs	
	as input.	as input.	
		1) Check that the number of ele-	pass
		number in the input 2) Check	
		that each output AO contains	
		that each output AO contains	
	Tests that the metropolis2D	Test that the metropolis2D	pass
05	method works with a mix of	method works with an input of	pass
	different shaped AOs as input.	matrices and vectors and single	
	r i i i i i r i r	AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	-
		input. 2) Check that each output	
		AO contains the correct data.	
06	Tests that the metropolis2D	Test that the result of applying	pass
00	method properly applies history.	the metropolis2D method can be	
I		processed back to an m-file.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/mcmc			
		1) Check that the last entry in	pass
		the history of 'm' corresponds	
		to 'metropolis2D'. 2) Check that	
		the re-built object is the same	
		object as 'm'. THIS IS NOT	
		TRUE FOR METROPOLIS BE-	
		CAUSE THE OUTPUT CHAIN	
		IS ONLY EQUAL STATISTI-	
		CALLY	
07	Tests that the metropolis2D	Test that the tfe method can	pass
07	method can not modify the	not modify the input AO. The	
	input AO.	method must throw an error for	
		the modifier call.	
		1) Nothing to check.	pass

Table 81: Unit tests for ao/mcmc.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/md5			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.9	Tests that the md5 method	Test that the md5 method works	pass
02	works with a vector of AOs as	for a vector of AOs as input.	
	input.	1) Check that the number of ele-	pass
	-	ments in 'out' are the same as in	
		'atvec' 2) Check that each output	
		AO contains the correct data.	
0.0	Tests that the md5 method	Tests that the md5 method	pass
03	works with a matrix of AOs as	works with a matrix of AOs as	1
	input.	input.	
	1	1) Check that the number of el-	pass
		ements in 'out' are the same as	1
		in 'atmat' 2) Check that each	
		output AO contains the correct	
		data	
	Tests that the md5 method	Tests that the md5 method	nass
04	works with a list of AOs as	works with a list of AOs as in-	Pass
	input	put	
	mput	1) Check that the number of el-	pass
		ements in 'out' are the same as	Pass
		in the input 2) Check that each	
		AO contains the correct	
		data	
	Tests that the md5 method	Tests that the md5 method	nase
05	works with a mix of different	works with a mix of different	pass
	shaped AOs as input	shaped AOs as input	
	shaped AOs as input.	1) Chock that the number of al	neg
		amonta in 'out' are the same as	pass
		in the input 2) Check that each	
		autput AQ contains the compact	
		data	
	Tests that the md5 method	Quala.	- Decc
06	rests that the hidd method	special case for one input be-	pass
	and not a coll of a string	cause in this case retruns indo a	
	and not a cell of a string	string and not a cell.	
		1) Uneck that the output is a	pass
		string 2) Check that each output	
		AO contains the correct data.	

Table 82: Unit tests for a o/md5.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mean			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually	pass
		1) Check that getInfo call re-	nass
		turned an minfo object in all	pass
		cases 2) Check that all plists	
		have the correct parameters	
	Tests that the [ao/mean]	Test that the [20/mean] method	nass
02	method works with a vector of	works for a vector of objects as	pass
	objects as input	input	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	Pass
		'vec' 2) Check that each output	
		object contains the correct data	
	Tests that the [ao/mean]	Test that the [ao/mean] method	pass
03	method works with a matrix of	works for a matrix of objects as	Pass
	objects as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	P
		'mat' 2) Check that each output	
		object contains the correct data.	
	Tests that the [ao/mean]	Test that the [ao/mean] method	pass
04	method works with a list of	works for a list of objects as in-	1
	objects as input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	-
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/mean]	Tests that the [ao/mean] method	pass
05	method works with a mix of	works with a mix of different ar-	
	different arrays of objects as	rays of objects as input.	
	input.	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
06	Tests that the [ao/mean]	Test that the result of apply-	pass
	method properly applies history.	ing the [ao/mean] method can be	
		processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[ao/mean]'. 2) Check that the	
		re-built object is the same object	
		as the input.	
07	Tests that the [ao/mean]	Test that the [ao/mean] method	pass
	method can modify the input	can modify the input object by	
	AO.	calling with no output and that	
		the method doesn't change the	
		input of the function notation	
		(with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/mean			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the [ao/mean]	
		value of the copy 4) Check that	
		out and amodi are the same	
100	Test that the [ao/mean] method	Test that the [ao/mean] method	pass
108	uses the plist to get the axis.	uses the plist to get the axis.	
	This is intended to test methods	1) Check that the [ao/mean]	pass
	like ao/mean and ao/std which	method applies to the x-axis	
	return different data types	2) Check that the [ao/mean]	
	depending on which axis is	method applies to the v-axis	
	selected.	3) Check that the [ao/mean]	
		method applies to both axes 4)	
		Check that the re-built object is	
		the same as in ' $out[13]$ '.	
	Test the shape of the data in	Test that the [ao/mean] method	pass
09	AOs.	keeps the data shape of the input	1
		object. The input AO data must	
		be an array with row data and/or	
		column data.	
		1) Check that the shape of the	pass
		data doesn't change.	P
10	Check that the [ao/mean]	Call the method with a list of	pass
10	method pass back the output	output variables and with a sin-	-
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	-
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/mean]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 83: Unit tests for a o/mean.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/median			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the [ao/median]	Test that the [ao/median]	pass
02	method works with a vector of	method works for a vector of	
	objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'vec' 2) Check that each output	
		object contains the correct data.	
03	Tests that the [ao/median]	Test that the [ao/median]	pass
	method works with a matrix of	method works for a matrix of	
	objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
04	Tests that the [ao/median]	Test that the [ao/median]	pass
04	method works with a list of	method works for a list of	
	objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/median]	Tests that the [ao/median]	pass
	method works with a mix of	method works with a mix of	
	different arrays of objects as	different arrays of objects as	
	input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
06	Tests that the [ao/median]	Test that the result of applying	pass
	method properly applies history.	the [ao/median] method can be	
		processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[ao/median]'. 2) Check that	
		the re-built object is the same	
		object as the input.	
07	Tests that the [ao/median]	Test that the [ao/median]	pass
	method can modify the input	method can modify the input	
	AO.	object by calling with no output	
		and that the method doesn't	
		change the input of the function	
		notation (with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/median			
		1) Check that 'out' and 'aeq'	pass
		are now different. 2) Check that	
		'aeq' is not changed 3) Check	
		that the modified input is the	
		[ao/median] value of the copy 4)	
		Check that out and amodi are	
		the same	
	Test that the [ao/median]	Test that the [ao/median]	pass
108	method uses the plist to get the	method uses the plist to get the	P
	axis This is intended to test	avis	
	methods like a /mean and	1) Check that the $\left[\frac{1}{20}\right]$	nase
	ac/std which roturn different	mothod applies to the vavis	pass
	data types depending on which	2 Check that the $\left[a_{0}/modian\right]$	
	avia is selected	method applies to the wavis	
	axis is selected.	$\begin{array}{c} \text{method} \text{applies} \text{to} \text{the } y \text{-axis} \\ 2 \end{array}$	
		5) Check that the [ao/median]	
		method applies to both axes 4)	
		Check that the re-built object is	
		the same as in $[out[13]]$.	
09	Test the shape of the data in	Test that the [ao/median]	pass
	AOs.	method keeps the data shape of	
		the input object. The input AO	
		data must be an array with row	
		data and/or column data.	
		1) Check that the shape of the	pass
		data doesn't change.	
10	Check that the [ao/median]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/median]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	-
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	T

Table 84: Unit tests for ao/median.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/min			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters	pass
02	Tests that the [ao/min] method works with a vector of objects as input.	Test that the [ao/min] method works for a vector of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'vec' 2) Check that each output object contains the correct data.	pass
03	Tests that the [ao/min] method works with a matrix of objects as input.	Test that the [ao/min] method works for a matrix of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
04	Tests that the [ao/min] method works with a list of objects as input.	Test that the [ao/min] method works for a list of objects as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
05	Tests that the [ao/min] method works with a mix of different arrays of objects as input.	Tests that the [ao/min] method works with a mix of different ar- rays of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
06	Tests that the [ao/min] method properly applies history.	Test that the result of applying the [ao/min] method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[ao/min]'. 2) Check that the re-built object is the same object as the input.	pass
07	Tests that the [ao/min] method can modify the input AO.	Test that the [ao/min] method can modify the input object by calling with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/min			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the [ao/min]	
		value of the copy 4) Check that	
		out and amodi are the same	
20.0	Test that the [ao/min] method	Test that the [ao/min] method	pass
208	uses the plist to get the axis.	uses the plist to get the axis.	
	This is intended to test methods	1) Check that the [ao/min]	pass
	like ao/max and ao/min which	method applies to the x-axis 2)	
	only allow 'x' and 'y' choices.	Check that the [ao/min] method	
		applies to the y-axis 3) Check	
		that the [ao/min] method applies	
		to both axes 4) Check that the	
		re-built object is the same as in	
		'out[13]'.	
00	Test the shape of the data in	Test that the [ao/min] method	pass
09	AOs.	keeps the data shape of the in-	
		put object. The input AO data	
		must be an array with row data	
		and/or column data.	
		1) Check that the shape of the	pass
		data doesn't change.	
10	Check that the [ao/min] method	Call the method with a list of	pass
10	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/min] method	Call the method with a list of	pass
11	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 85: Unit tests for a o/min.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
rule1 (tsdata and tsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and fsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/minus			
rule1 (cdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (tsdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (tsdata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (cdata and tsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule1 (xydata and	Tests the arithmetic operators	Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
tsdata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
fsdata)	rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and fsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule1 (cdata and	Tests the arithmetic operators	Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	L
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single tsdata and vector tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector tsdata and single tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rulo2 (cinglo	Tasts the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata and vector tsdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector tsdata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single cdata and vector tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule2 (vector tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and single cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single fsdata and vector fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector fsdata and single fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single xydata and vector fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector fsdata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single cdata and vector fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule2 (vector fedata	Tasts the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and single cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single xydata and vector xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector xydata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single cdata and vector xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector xydata and single cdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector cdata and single cdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single cdata and vector cdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule ³ (vector tsdata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rulo3 (voctor edata	Tasts the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector tsdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule3 (vector fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule3 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector fsdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule3 (vector	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata and vector cdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector cdata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector cdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule3 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector cdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	I
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector tsdata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector fsdata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector fsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector xydata and vector xydata)	rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector xydata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/minus			
rulo4 (voctor edata	Tasts the arithmetic operators	Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector cdata)	rule 4.	rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP tsdata and single tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single tsdata and NxP tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule5 (NxP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and single xydata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP tsdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule5 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP fsdata and single fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single fsdata and NxP fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule5 (NyP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and single xydata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP fsdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rulo5 (single edata	Tasts the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP xydata and single xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule5 (NxP xvdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and single cdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single cdata and NxP xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP cdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule5 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP cdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule6 (NxP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rulo6 (vostor sdata	Tasts the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule6 (NyP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule6 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP fsdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule6 (NyP yudata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector cdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule6 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP cdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule7 (NxP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule7 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP tsdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule7 (NxP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule7 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP fsdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule7 (NxP xvdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector cdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	1
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule7 (vector cdata	Tasts the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP cdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
mula@ (NuD tadata	Tests the arithmetic encretors	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule8 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule8 (NxP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule8 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule8 (NyP yvdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector cdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	разы
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule8 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP cdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	равя
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP tsdata and NxQ tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ tsdata and NxP tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule9 (NxP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxQ xydata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	равя
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP tsdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rulo0 (NrO adata	Tosts the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP fsdata and NxQ fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ fsdata and NxP fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule9 (NxP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxQ xydata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP fsdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule9 (NxQ cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP fsdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP xydata and NxQ xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rulo0 (NyP yydata	Tosts the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxQ cdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ cdata and NxP xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP cdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule9 (NxQ cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP cdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	L
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rulo10 (NyP edata	Tasts the arithmetic energy	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
mula10 (NyP adota	Tests the arithmetic encounters	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
rule10 (NxP	Tests the arithmetic operators	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
xydata and NxP cdata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	1
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/minus			
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
minus			pass
tosta (fadata and	Tests all arithmatic anonations	Tests all arithmatic anomations	pass
tsdata)	which are not allowed.	which are not allowed.	pass
		Here we test some negative cases.	pass
tests (tsdata and fsdata)	Tests all arithmetic operations which are not allowed.	Tests all arithmetic operations which are not allowed.	pass
		Here we test some negative cases.	pass
tests (AO no data and tsdata)	Tests all arithmetic operations which are not allowed.	Tests all arithmetic operations which are not allowed.	pass
	Trata all anithmatic an anationa	There we test some negative cases.	pass
AO no data)	which are not allowed.	which are not allowed.	pass
		Here we test some negative cases.	pass
tsdata)	which are not allowed.	which are not allowed.	pass
		Here we test some negative cases.	pass
tests (different x	1ests all arithmetic operations	lests all arithmetic operations	pass
units in tsdata)	which are not allowed.	which are not allowed.Here we test some negative cases.	pass
tests (different x	Tests all arithmetic operations	Tests all arithmetic operations	pass
values in fsdata)	which are not allowed.	which are not allowed.	
		Here we test some negative cases.	pass
tests (negative	Tests all arithmetic operations	Tests all arithmetic operations	pass
test)	which are not allowed.	which are not allowed.	
		Here we test some negative cases.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/minus			
tests (negative	Tests all arithmetic operations	Tests all arithmetic operations	pass
test)	which are not allowed.	which are not allowed.	
		Here we test some negative cases.	pass

Table 86: Unit tests for ao/minus.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mode			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set	pass
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the [ao/mode]	Test that the [ao/mode] method	pass
	method works with a vector of	works for a vector of objects as	
	objects as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'vec' 2) Check that each output	
		object contains the correct data.	
03	Tests that the [ao/mode]	Test that the [ao/mode] method	pass
	method works with a matrix of	works for a matrix of objects as	
	objects as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
04	Tests that the [ao/mode]	Test that the [ao/mode] method	pass
01	method works with a list of	works for a list of objects as in-	
	objects as input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/mode]	Tests that the [ao/mode] method	pass
	method works with a mix of	works with a mix of different ar-	
	different arrays of objects as	rays of objects as input.	
	input.	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
06	Tests that the [ao/mode]	Test that the result of apply-	pass
	method properly applies history.	ing the [ao/mode] method can be	
		processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[ao/mode]'. 2) Check that the	
		re-built object is the same object	
		as the input.	
07	1 lests that the [ao/mode]	[lest that the [ao/mode] method	pass
	method can modify the input	can modify the input object by	
	AO.	calling with no output and that	
		the method doesn't change the	
		input of the function notation	
		(with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/mode			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the [ao/mode]	
		value of the copy 4) Check that	
		out and amodi are the same	
100	Test that the [ao/mode] method	Test that the [ao/mode] method	pass
108	uses the plist to get the axis.	uses the plist to get the axis.	
	This is intended to test methods	1) Check that the [ao/mode]	pass
	like ao/mean and ao/std which	method applies to the x-axis	-
	return different data types	2) Check that the [ao/mode]	
	depending on which axis is	method applies to the y-axis	
	selected.	3) Check that the [ao/mode]	
		method applies to both axes 4)	
		Check that the re-built object is	
		the same as in ' $out[13]$ '.	
	Test the shape of the data in	Test that the [ao/mode] method	pass
09	AOs.	keeps the data shape of the input	-
		object. The input AO data must	
		be an array with row data and/or	
		column data.	
		1) Check that the shape of the	pass
		data doesn't change.	-
10	Check that the [ao/mode]	Call the method with a list of	pass
10	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/mode]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 87: Unit tests for a o/mode.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mpower			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the mpower method works with a vector of AOs as input.	Test that the mpower method works for a vector of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is 1 2) Check that each output AO contains the cor- rect data. 3) Check the y-units	pass
03	Tests that the mpower method works with a matrix of AOs as input.	Test that the mpower method works for a matrix of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is 1 2) Check that each output AO contains the cor- rect data.	pass
04	Tests that the mpower method works with a list of AOs as input	Test that the mpower method works for a list of AOs as input.	pass
	input.	ments in 'out' is 1. 2) Check that each output AO contains the cor- rect data. 3) Check the y-units	pass
05	Tests that the mpower method works with a mix of different shaped AOs as input.	Test that the mpower method works with an input of matrices and vectors and single AOs.	pass
06	Tests that the mpower method properly applies history.	Test that the result of applying the mpower method can be pro- cessed back to an m-file.	pass
		 Check that the last entry in the history of 'out1' and 'out2' corresponds to 'mpower'. Check that the re-built objects are the same objectd as 'out1' and 'out2'. 	pass
07	Tests that the mpower method can modify the input AO.	Test that the mpower method can modify the input AO by call- ing with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass
		1) Check that 'at4' and 'ain' are now different. 2) Check that 'ain' is mpower(at4).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mpower			
08	Test the shape of the output.	Test that the mpower method keeps the data shape of the in- put object. The input AO must be an AO with row data and an	pass
		AO with column data.	
		1) Nothinf to check	pass

Table 88: Unit tests for a o/mpower.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
rule1 (tsdata and tsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and fsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule1 (cdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (tsdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (tsdata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (cdata and tsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule1 (xydata and	Tests the arithmetic operators	Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
tsdata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule1 (cdata and	Tests the arithmetic operators	Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
fsdata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and fsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rulo1 (adote and	Tasts the arithmetic energy	Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single tsdata and vector tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector tsdata and single tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule2 (single	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata and vector tsdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	P
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector tsdata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single cdata and vector tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule2 (vector tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and single cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single fsdata and vector fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector fsdata and single fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rulo2 (single	Tasts the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata and vector fsdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	равы
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector fsdata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single cdata and vector fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule2 (vector fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and single cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single xydata and vector xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector xydata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule2 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	1
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector xydata and single cdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector cdata and single cdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule2 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
mile? (vector tedato	Trate the orithmetic energy of	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
mile? (vector edote	Tasts the arithmetic encretors	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector tsdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
mula? (vactor fadata	Tasts the arithmetic encretors	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule3 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector fsdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector cdata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector cdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rulo3 (voctor edata	Tosts the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector cdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule4 (vector tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 4.	rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	I
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector fsdata	Tests the arithmetic operators	Tests the arithmetic operators	pass
and vector isdata)	rule 4.	rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector fsdata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata	Tests the arithmetic operators	Tests the arithmetic operators	pass
and vector fsdata)	rule 4.	rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/mrdivide			
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector	Tests the arithmetic operators	Tests the arithmetic operators	pass
xydata and vector xydata)	rule 4.	rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector xydata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata	Tests the arithmetic operators	Tests the arithmetic operators	pass
and vector xydata)	rule 4.	rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule4 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector cdata)	rule 4.	rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP tsdata and single tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single tsdata and NxP tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
wlo5 (NyD todata	Tests the emithmetic encounters	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and single xydata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP tsdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule5 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP tsdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP fsdata and single fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single fsdata and NxP fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
milet (NrD fadata	Tests the orithmetic energy of	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and single xydata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP fsdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rulo5 (single edate	Tests the prithmatic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP xydata and single xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
mile5 (NuD midete	Tests the emithmetic encounters	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and single cdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single cdata and NxP xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP cdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rulo5 (single edete	Tasts the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP cdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
mulof (NuD tadata	Tests the orithmetic energy of	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
mlef (meeter edete	Trate the emithmetic encoders	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
mulof (NuD fodata	Tests the emithmetic encounters	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rulo6 (voctor edata	Tasts the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP xydata	Tests the arithmetic operators	Tests the arithmetic operators	pass
and vector cdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP cdata	Tests the arithmetic operators	Tests the arithmetic operators	pass
and vector cdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
mile6 (verter edete	Tosts the arithmetic encreters	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP cdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule7 (NxP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector xydata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rulo7 (voctor edata	Tosts the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule7 (NxP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector xydata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule7 (vector cdata and NxP fsdata)	Tests the arithmetic operators rule 7.	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators rule 7 for each data type: xydata.	pass
,		fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule7 (NxP xydata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rulo7 (vostor adota	Tasts the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP cdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule8 (NxP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector xydata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	-
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule8 (vector cdata	Tasts the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
mulos (NuD fodata	Tests the emithmetic encounters	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rulos (voctor cdata	Tasts the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
mile ⁸ (NuD midete	Tests the emithmetic encounters	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector cdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
mula? (uppeton adota	Tasts the exithmetic encreters	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP cdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP tsdata and NxQ tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ tsdata and NxP tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxQ xydata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP tsdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP fsdata and NxQ fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ fsdata and NxP fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
	Trata the suither time means to a	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxQ xydata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP fsdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ cdata and NxP fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP xydata and NxQ xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP xydata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ cdata and NxP xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP cdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
mile0 (NrO edete	Trate the exit breating operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP cdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rula10 (NyP tadata	Tasts the arithmetic operators	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP xydata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rulo10 (NyP fsdata	Tosts the arithmetic operators	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP xydata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule10 (NxP cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP fsdata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
rule10 (NyP	Tests the arithmetic operators	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata and NxP cdata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/mrdivide			
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rdivide			pass
			pass
tsdata)	which are not allowed.	which are not allowed.	pass
		Here we test some negative cases.	pass
tests (tsdata and fsdata)	Tests all arithmetic operations which are not allowed.	Tests all arithmetic operations which are not allowed.	pass
		Here we test some negative cases.	pass
tests (AO no data and tsdata)	Tests all arithmetic operations which are not allowed.	Tests all arithmetic operations which are not allowed.	pass
	Trata all anithmatic an anationa	There we test some negative cases.	pass
AO no data)	which are not allowed.	which are not allowed.	pass
		Here we test some negative cases.	pass
tsdata)	which are not allowed.	which are not allowed.	pass
		Here we test some negative cases.	pass
tests (different x	lests all arithmetic operations	lests all arithmetic operations	pass
values in fsdata)	which are not allowed.	which are not allowed. Here we test some negative cases.	pass
tests (negative	Tests all arithmetic operations	Tests all arithmetic operations	pass
test)	which are not allowed.	which are not allowed.	-
		nere we test some negative cases.	pass
tests (negative	lests all arithmetic operations	lests all arithmetic operations	pass
test)	which are not allowed.	which are not allowed.	
		Here we test some negative cases.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/	mrc	liv	ide
/			

Table 89: Unit tests for ao/mrdivide.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/mtimes			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.4	Tests that the mtimes method	Test that the mtimes method	pass
04	works with a list of AOs as	works for a list of AOs as input.	
	input.	1) Check that the number of ele-	pass
		ments in 'out' is 1. 2) Check that	-
		each output AO contains the cor-	
		rect data. 3) Check the v-units	
	Tests that the mtimes method	Test that the result of apply-	pass
06	properly applies history.	ing the mtimes method can be	1
		processed back with the rebuild	
		method.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	1
		to 'mtimes'. 2) Check that the re-	
		built object is the same object as	
		'out'.	
07	Tests that the mtimes method	Tests that the mtimes method	pass
07	can not be used as a modifier	can not be used as a modifier	1
	method.	method. The command should	
		fail.	
		1) Nothing to test.	pass
00		Test that the mtimes method	pass
08	Test the shape of the output.	keeps the data shape of the in-	-
		put object. The input AO must	
		be an AO with row data and an	
		AO with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
00	Test the method with all data	Test that the mtimes method	pass
09	objects.	works with cdata-, fsdata-,	
		tsdata-, and xydata objects	
		1) Check that the shpe of the	pass
		data doesn't change. 2) Check	
		that re-building of output is the	
		same as the output	

Table 90: Unit tests for ao/mtimes.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/ne			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minto object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the ne method works	Test that the ne method works	pass
	with a vector of AOs as input.	for a vector of AOs as input.	
		Test the positive and the nega-	
		tive case.	
		1) Check the output of the ne	pass
		function.	
03	Tests that the ne method works	Test that the ne method works	pass
	with a matrix of AOs as input.	for a matrix of AOs as input.	
		Test the positive and the nega-	
		tive case.	
		1) Check the output of the ne	pass
		function.	
04	Tests that the ne method works	The ne method doesn't works for	pass
	with a list of AOs as input.	a list of AOs as input. Nothing to	
		do.	
			pass
05	Tests that the ne method works	The ne method doesn't works for	pass
	with a mix of different shaped	a list of AOs as input. Nothing to	
	AOs as input.	do.	
			pass
06	Tests that the ne method	The ne method doesn't change	pass
	properly applies history.	the AO, thus will no history	
		added. Nothing to do	
			pass
07	Test the ne method with an	Test the ne method with the ex-	pass
	exception list. The function	ception 'name'. Use the option	
	ao/ne use the function ao/eq so	'internal' to suppress the history.	
	It is not necessary to check all	It is necessary to add 'created' to	
	possibilities of the exception list.	the exception list because aa is	
		created at an other time.	
		1) Check the output.	pass
08	Test the ne method with an	Test that the ne method uses the	pass
	exception list which is in a plist.	exception list in a plist.	
		1) Check the output.	pass

Table 91: Unit tests for ao/ne.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/noisegen1D			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set	pass
		1) Check that get info call no	noga
		1) Check that gethio can re-	pass
		cases 2) Check that all plists	
		cases. 2) Check that all plists	
	Tests that the noiseren1D	Test that the neisegran1D method	negg
02	method works with a voctor of	works for a vector of AOs as in	pass
	And as input	works for a vector of AOS as in-	
	AOS as input.	1) Check that the number of el	nase
		amonta in 'out' is the same as in	pass
		'and 2) Check that each output	
		AO contains the correct data	
	Tests that the noisesen1D	The contains the confect data.	naga
03	method works with a matrix of	works for a matrix of AOs as in	pass
	And a simplet	works for a matrix of AOS as m-	
	AOS as input.	1) Check that the number of al	neg
		amonta in 'out' is the same as	pass
		in 'atmat' 2) Chock that each	
		$\frac{111}{111}$ at $\frac{111}{2}$ $\frac{111}{111}$ $\frac{111}{2}$ $\frac{111}{111}$ $\frac{1111}{1111}$ $\frac{111}{1111}$ $\frac{1111}{1111}$ $\frac{1111}{11111}$ $\frac{1111}{1111}$ $\frac{1111}{11111}$ $\frac{1111}{11111}$ $\frac{1111}{11111}$ $\frac{1111}{11111}$ $\frac{1111}{11111}$ $\frac{1111}{11111}$ $\frac{1111}{11111}$ $\frac{1111}{11111}$ $\frac{1111}{111111}$ $\frac{11111}{111111}$ $\frac{11111}{1111111}$ $\frac{11111}{111111111111111111111111111111$	
		data	
	Tests that the noiseren1D	Test that the noise ren1D method	paga
04	method works with a list of AOs	works for a list of AOs as input	pass
	as input	1) Check that the number of al	nase
	as input.	amonts in 'out' is the same as in	pass
		input 2) Check that each output	
		$\Delta \Omega$ contains the correct data	
	Tosts that the noisegon1D	To contains the correct data.	nase
05	method works with a mix of	works with an input of matrices	pass
	different shaped AOs as input	and vectors and single AOs	
	different shaped rios as input.	1) Check that the number of el-	nass
		ements in 'out' is the same as in	pass
		input 2) Check that each output	
		AQ contains the correct data	
	Tests that the noisegen1D	Test that the result of applying	nass
06	method properly applies history	the noisegen1D method can be	pass
	meened property apprese metory.	processed back	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	pass
		to 'noisegen1D'. 2) Check that	
		the re-built object is the same	
		object as the input.	
	Tests that the noisegen1D	Test that the noisegen1D method	pass
07	method can modify the input	can modify the input AO by call-	Pass
	AO.	ing with no output and that the	
		method doesn't change the input	
		of the function notation (with a	
		equal sign).	


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/noisegen1D			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is changed	
		Test that the noisegen1D method	pass
08	Test the shape of the output.	keeps the data shape of the in-	P
		put object. The input AO must	
		be an AO with row data and an	
		A with column data	
		1) Check that the shape of the	naga
		data dagan't shanga	pass
		data doesn't change.	
09	Check that the noisegen1D	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
10	Tests that the noisegen1D	Test that the result of applying	pass
10	method properly applies history	the noisegen1D method can be	-
	for set filter.	processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	-
		to 'noisegen1D'. 2) Check that	
		the re-built object is the same	
		object as the input.	
	Tests that the noisegen1D	Test that the noisegen1D method	pass
11	method can modify the input	can modify the input AO by call-	1
	AO for set 'filter'	ing with no output and that the	
		method doesn't change the input	
		of the function notation (with a	
		equal sign)	
		1) Check that 'out' and 'and 'and	nass
		now different 2) Check that 'and'	pass
		is not changed	
	Check that the noisegon1D	Call the mothod with a list of	nagg
12	method pass back the output	output variables and with a sin	pass
	objects to a list of output	ale output variable. Additionaly	
	variables or to a single variable	shock that the rebuild method	
	variables of to a single variable.	works on the output	
		1) Check that the output.	negg
		toing the right number of -1	pass
		tains the right number of OD-	
		Jects 2) Uneck that the rebuild	
		method produces the same ob-	
		ject as 'out'.	

Table 92: Unit tests for ao/noisegen1D.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/noisegen2D			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	tor no sets, all sets, and each set	
		1) Chock that got Info call ro	negg
		turned an minfo object in all	pass
		cases 2) Check that all plists	
		have the correct parameters	
	Tests that the noisegen2D	Test that the noisegen 2D method	nass
02	method works with a vector of	works for a vector of AOs as in-	pass
	AOs as input	put	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	Pabb
		'acv2' 2) Check that each output	
		AO contains the correct data.	
0.2	Tests that the noisegen2D	Test that the noisegen2D method	pass
03	method works with a matrix of	works for a matrix of AOs as in-	
	AOs as input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as	
		in 'atmat' 2) Check that each	
		output AO contains the correct	
		data.	
04	Tests that the noisegen2D	Test that the noisegen2D method	pass
01	method works with a list of AOs	works for a list of AOs as input.	
	as input.	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
	Trate that the main new 2D	AU contains the correct data.	
05	method works with a mix of	rest that the whitehild method	pass
	different shaped AOs as input	and vectors and single AOs	
	different shaped AOs as input.	1) Check that the number of el-	nass
		ements in 'out' is the same as in	Pass
		input, 2) Check that each output	
		AO contains the correct data.	
	Tests that the noisegen2D	Test that the result of applying	pass
06	method properly applies history.	the noisegen2D method can be	
		processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'noisegen2D'. 2) Check that	
		the re-built object is the same	
		object as the input.	
07	The noisegen2D method can not	The noisegen2D method can not	pass
	modify the input AO.	modify the input AO.	
		1) Nothing to do.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/noisegen2D			
08	Test the shape of the output.	Test that the noisegen2D method keeps the data shape of the input	pass
		object. The input AO must be a	
		couple AO with row data and a	
		couple AO with column data.	
		1) Check that the shape of the	pass
		data doesn't change.	
00	Check that the noisegen2D	Call the method with a list of	pass
03	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
10	Tests that the noisegen2D	Test that the result of applying	pass
10	method properly applies history.	the noisegen2D method can be	
		processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'noisegen2D'. 2) Check that	
		the re-built object is the same	
		object as the input.	

Table 93: Unit tests for ao/noisegen2D.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/norm			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the norm method works with a vector of AOs as input.	Test that the norm method works for a vector of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atvec' 2) Check that each output AO contains the correct data.	pass
03	Tests that the norm method works with a matrix of AOs as input.	Test that the norm method works for a matrix of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atmat' 2) Check that each output AO contains the correct data.	pass
04	Tests that the norm method works with a list of AOs as input.	Test that the norm method works for a list of AOs as input. 1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass pass
05	Tests that the norm method works with a mix of different shaped AOs as input.	Test that the norm method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the norm method properly applies history.	Test that the result of applying the norm method can be pro- cessed back to an m-file.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'norm'. 2) Check that the re- built object is the same object as 'out'.	pass
07	Tests that the norm method can modify the input AO.	Test that the norm method can modify the input AO by call- ing with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/norm			
		1) Check that 'at4' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		is norm(at4).	
08	Control the method with a plict	Test that the norm method can	pass
00	Control the method with a plist.	modify the single axis controlled	
		by the plist and the resuld can be	
		processed back to an m-file.	
		1) Check that the norm method	pass
		applies with different options 2)	
		Check that the re-built objects	
		are the same object as ' $out[14]$ '.	
00	Control the method with a plict	Test that the norm method keeps	pass
09	Control the method with a plist.	the data shape of the input ob-	
		ject. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	

Table 94: Unit tests for a o/norm.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/offset			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the offset method works with a vector of AOs as input.	Test that the offset method works for a vector of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atvec' 2) Check that each output AO contains the correct data.	pass
03	Tests that the offset method works with a matrix of AOs as input.	Test that the offset method works for a matrix of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atmat' 2) Check that each output AO contains the correct data.	pass
04	Tests that the offset method works with a list of AOs as input.	Test that the offset method works for a list of AOs as input. 1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
05	Tests that the offset method works with a mix of different shaped AOs as input.	Test that the offset method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the offset method properly applies history.	Test that the result of applying the offset method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'offset'. 2) Check that the re- built object is the same object as the input.	pass
07	Tests that the offset method can modify the input AO.	Test that the offset method can modify the input AO by call- ing with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/offset			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified have an offset of 3 4)	
		Check that out and amodi are	
		the same	
00	Test that the offset method uses	Test that the offset method uses	pass
08	the offset in a plist.	the offset in a plist.	-
	-	1) Check that the offset uses the	pass
		offset in the plist 2) Check that	-
		the re-built object is the same as	
		'out'	
00		Test that the offset method keeps	pass
09	Test the snape of the output.	the data shape of the input ob-	
		ject. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shape of the	pass
		data doesn't change.	
10	Check that the offset method	Call the method with a list of	pass
10	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/offset]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 95: Unit tests for a o/offset.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
rule1 (tsdata and tsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and fsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule1 (cdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (tsdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (tsdata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (cdata and tsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule1 (xydata and	Tests the arithmetic operators	Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
tsdata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule1 (cdata and	Tests the arithmetic operators	Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
fsdata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and fsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule1 (cdata and	Tests the arithmetic operators	Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
xydata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single tsdata and vector tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector tsdata and single tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
mile? (single	Tasts the exithmatic encretors	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata and vector tsdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector tsdata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single cdata and vector tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule2 (vector tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and single cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single fsdata and vector fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector fsdata and single fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rulo2 (single	Tasts the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata and vector fsdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector fsdata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single cdata and vector fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule2 (vector fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and single cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	I
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single xydata and vector xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector xydata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule2 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector xydata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector xydata and single cdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector cdata and single cdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule2 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule3 (vector tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule3 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector tsdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rulo3 (voctor fedata	Tosts the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule3 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector fsdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
mule2 (vestor	Tests the arithmetic encounters	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata and vector cdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector cdata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector cdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule3 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector cdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector fsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector fsdata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector fsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
mulat (reator	Tests the arithmetic encounters	Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata and vector xydata)	rule 4.	rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector xydata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule4 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector cdata)	rule 4.	rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP tsdata and single tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single tsdata and NxP tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and single xydata)	rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP tsdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule5 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP fsdata and single fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single fsdata and NxP fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule5 (NxP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and single xydata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP fsdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule5 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP fsdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	1
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP xydata and single xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule5 (NxP xvdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and single cdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single cdata and NxP xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP cdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule5 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP cdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule6 (NyP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule6 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP tsdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	I
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rulo6 (NyP fsdata	Tasts the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule6 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP fsdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	1
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule6 (NxP xvdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector cdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule6 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP cdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule7 (NxP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule7 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule7 (NxP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector xydata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule7 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP fsdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule7 (NxP xydata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule7 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP cdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule8 (NxP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	P
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule8 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	I man
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
mula? (NuD fadata	Tests the emithmetic encounters	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule8 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP fsdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	-
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule8 (NxP xvdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule8 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP cdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	1
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP tsdata and NxQ tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ tsdata and NxP tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
wildo (NyP todata	Tests the arithmetic encretors	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxQ xydata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP tsdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule9 (NxQ cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP fsdata and NxQ fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ fsdata and NxP fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
mule0 (NuP fedete	Tasts the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxQ xydata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP fsdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule9 (NxQ cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP xydata and NxQ xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule9 (NxP yydata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxQ cdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	равя
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ cdata and NxP xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP cdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule9 (NyQ cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP cdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule10 (NxP cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP tsdata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule10 (NxP cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP fsdata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
rule10 (NxP	Tests the arithmetic operators	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
xydata and NxP cdata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or			
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata	Tests the arithmetic operators	Tests the arithmetic operators	pass
and NxP cdata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
or			pass
(0.1			pass
tests (fsdata and	Tests all arithmetic operations	Tests all arithmetic operations	pass
tsdata)	which are not allowed.	which are not allowed.	
		Here we test some negative cases.	pass
fsdata)	which are not allowed.	which are not allowed.	pass
		Here we test some negative cases.	pass
tests (AO no data and tsdata)	Tests all arithmetic operations which are not allowed.	Tests all arithmetic operations which are not allowed.	pass
		Here we test some negative cases.	pass
AO no data)	which are not allowed.	which are not allowed.	pass
		Here we test some negative cases.	pass
tests (different fs in tsdata)	Tests all arithmetic operations which are not allowed.	Tests all arithmetic operations which are not allowed.	pass
(1. 2		Here we test some negative cases.	pass
tests (different x	Tests all arithmetic operations	Tests all arithmetic operations	pass
values in fsdata)	which are not allowed.	which are not allowed. Here we test some negative cases.	pass
tests (negative	Tests all arithmetic operations	Tests all arithmetic operations	pass
test)	which are not allowed.	which are not allowed.	Pass
		Here we test some negative cases.	pass
tests (negative	Tests all arithmetic operations	Tests all arithmetic operations	pass
test)	which are not allowed.	which are not allowed.	
		Here we test some negative cases.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/or

Table 96: Unit tests for ao/or.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/phase			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all	pass
		cases. 2) Check that all plists have the correct parameters.	
02	Tests that the [ao/phase] method works with a vector of objects as input.	Test that the [ao/phase] method works for a vector of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'vec' 2) Check that each output	pass
03	Tests that the [ao/phase] method works with a matrix of	Test that the [ao/phase] method works for a matrix of objects as	pass
	objects as input.	 1) Check that the number of elements in 'out' is the same as in 'mat' 2) Check that each output 	pass
04	Tests that the [ao/phase] method works with a list of objects as input	object contains the correct data. Test that the [ao/phase] method works for a list of objects as in-	pass
		 Check that the number of elements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data. 	pass
05	Tests that the [ao/phase] method works with a mix of different arrays of objects as	Tests that the [ao/phase] method works with a mix of different ar- rays of objects as input.	pass
	input.	1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
06	Tests that the [ao/phase] method properly applies history.	Test that the result of applying the [ao/phase] method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[ao/phase]'. 2) Check that the re-built object is the same object as the input.	pass
07	Tests that the [ao/phase] method can modify the input AO.	Test that the [ao/phase] method can modify the input object by calling with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/phase			
, <u>-</u>		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the [ao/phase]	
		value of the copy 4) Check that	
		out and amodi are the same	
00	Test the shape of the data in	Test that the [ao/phase] method	pass
09	AOs.	keeps the data shape of the input	-
		object. The input AO data must	
		be an array with row data and/or	
		column data.	
		1) Check that the shape of the	pass
		data doesn't change.	-
10	Check that the [ao/phase]	Call the method with a list of	pass
10	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/phase]	Call the method with a list of	pass
11	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 97: Unit tests for ao/phase.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
rule1 (tsdata and tsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and fsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule1 (cdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (tsdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (tsdata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (cdata and tsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule1 (xydata and	Tests the arithmetic operators	Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
tsdata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule1 (cdata and	Tests the arithmetic operators	Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
fsdata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and fsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule1 (cdata and	Tests the arithmetic operators	Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
xydata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single tsdata and vector tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector tsdata and single tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule2 (single	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata and vector tsdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	Pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector tsdata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single cdata and vector tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule2 (vector tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and single cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single fsdata and vector fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector fsdata and single fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single xydata and vector fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector fsdata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single cdata and vector fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule2 (vector fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and single cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single xydata and vector xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector xydata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single cdata and vector xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector xydata and single cdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector cdata and single cdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rulo2 (single edate	Tosts the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule3 (vector tsdata	Tasts the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule3 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector tsdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	-
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule3 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector fsdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rulo2 (vostor	Tosts the arithmetic energy	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata and vector cdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector cdata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector cdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule3 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule4 (vector tsdata	Tasts the arithmetic operators	Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 4.	rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector fsdata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector fsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector xydata and vector xydata)	rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector xydata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector cdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP tsdata and single tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single tsdata and NxP tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP tsdata and single xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP tsdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule5 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP tsdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP fsdata and single fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single fsdata and NxP fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and single xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP fsdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule5 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP xydata and single xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule5 (NxP xvdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and single cdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single cdata and NxP xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP cdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule5 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP cdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule6 (NxP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	разэ
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule6 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP tsdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule6 (NxP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	P
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule6 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP fsdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	1
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule6 (NxP xvdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule6 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP cdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule7 (NyP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule7 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP tsdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
mulo7 (NuD fadata	Tasts the exithmetic encretors	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule7 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	1
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule7 (NxP xvdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	-
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule7 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP cdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule8 (NyP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule8 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rulos (NyP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rulos (vostor adata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP xydata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule8 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP cdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP tsdata and NxQ tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ tsdata and NxP tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule9 (NyP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxQ xydata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	равя
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP tsdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule9 (NxQ cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP fsdata and NxQ fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ fsdata and NxP fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
mule0 (Na P. fadata	Tasts the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxQ xydata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP fsdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule9 (NxQ cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP xydata and NxQ xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
mile0 (NeD mideto	Trate the orithmetic energy and	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxQ cdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ cdata and NxP xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP cdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule9 (NxQ cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP cdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule10 (NyP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NyP yvdata)	rule 10	rule 10 for each data type: yy-	pass
		data, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata	Tests the arithmetic operators	Tests the arithmetic operators	pass
and NxP cdata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule10 (NxP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP xydata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	Pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule10 (NxP cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP fsdata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
rule10 (NxP	Tests the arithmetic operators	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
xydata and NxP cdata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	I man
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
plus			pass
tosta (fadoto ond	Tests all arithmatic anonations	Tests all arithmatic anomations	pass
tsdata)	which are not allowed.	which are not allowed.	pass
		Here we test some negative cases.	pass
tests (tsdata and fsdata)	Tests all arithmetic operations which are not allowed.	Tests all arithmetic operations which are not allowed.	pass
(1.0.1		Here we test some negative cases.	pass
and tsdata)	Tests all arithmetic operations which are not allowed.	Tests all arithmetic operations which are not allowed.	pass
tests (tadata and	Tests all arithmatic an anations	Tests all arithmetic anonations	pass
AO no data)	which are not allowed.	which are not allowed.	pass
		Here we test some negative cases.	pass
tests (different is in tsdata)	which are not allowed.	Tests all arithmetic operations which are not allowed.	pass
		Here we test some negative cases.	pass
tests (different x	Iests all arithmetic operations	lests all arithmetic operations	pass
units in tsdata)	which are not allowed.	which are not allowed.Here we test some negative cases.	pass
tests (different x	Tests all arithmetic operations	Tests all arithmetic operations	pass
values in fsdata)	which are not allowed.	which are not allowed.	
		Here we test some negative cases.	pass
tests (negative	Tests all arithmetic operations	Tests all arithmetic operations	pass
test)	which are not allowed.	which are not allowed.	
		Here we test some negative cases.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/plus			
tests (negative	Tests all arithmetic operations	Tests all arithmetic operations	pass
test)	which are not allowed.	which are not allowed.	
		Here we test some negative cases.	pass

Table 98: Unit tests for ao/plus.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/polyfit			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the polyfit method fails with with a vector of AOs as input.	Test that the polyfit method works with a vector of AOs as in- put.	pass
		1) Check we have the correct number of output objects in the version with single output 2) Check we have the correct num- ber of elements in the output ob- jects	pass
03	Tests that the polyfit method works with a matrix of AOs as input.	Test that the polyfit method works for a matrix of AOs as input.1) Check we have the correct	pass pass
04	Tests that the polyfit method works with a list of AOs as	number of output objects Test that the polyfit method works for a list of AOs as input.	pass
	input.	1) Check we have the correct number of output objects in the version with single output 2) Check we have the correct num- ber of elements in the output ob- jects	pass
05	Tests that the polyfit method works with a mix of different shaped AOs as input.	Test that the polyfit method works with an input of matrices and vectors and single AOs. 1) Check we have the correct	pass pass
06	Tests that the polyfit method properly applies history.	Test that the result of applying the polyfit method can be pro- cessed back to an m-file.	pass
		1) Check that the last entry in the history of 'out1' corresponds to 'polyfit'. 2) Check that the re- built object is the same object as 'out'.	pass
07	Tests that the polyfit method cannot modify the input AO.	Test that the polyfit method can- not modify the input AO by call- ing with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass
		I I INOUTING TO CHECK	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/polyfit			
00	Tests that the polyfit method	Tests that the polyfit method	pass
08	handles units correctly.	handles units correctly.	
		1) Check that the units in both	pass
		cases are yunits/(xunits) (j)	
00	Tests that the ao/polyfit	Test that applying polyfit works	pass
09	method agrees with MATLAB's	on a single AO.	
	polyfit when configured to use	1) Check that output agrees with	pass
	the same parameters.	the output of MATLAB's polyfit.	

Table 99: Unit tests for ao/polyfit.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/power			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set	pass
		 Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters. 	pass
02	Tests that the power method works with a vector of AOs as input.	Test that the power method works for a vector of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is 1 2) Check that each output AO contains the cor- rect data.	pass
03	Tests that the power method works with a matrix of AOs as input.	Test that the power method works for a matrix of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is 1 2) Check that each output AO contains the cor- rect data.	pass
04	Tests that the power method works with a list of AOs as input	Test that the power method works for a list of AOs as input.	pass
	input.	ments in 'out' is 1. 2) Check that each output AO contains the cor- rect data. 3) Check the y-units	Pabb
05	Tests that the power method works with a mix of different shaped AOs as input.	Test that the power method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of ele- ments in 'out' is 1. 2) Check that each output AO contains the cor- rect data.	pass
06	Tests that the power method properly applies history.	Test that the result of applying the power method can be pro- cessed back to an m-file.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'power'. 2) Check that the re- built object is the same object as 'out'.	pass
07	Tests that the power method can modify the input AO.	Test that the power method can modify the input AO by call- ing with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/power			
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		is power(at1).	
08	Test the shape of the output	Test that the power method	pass
08	rest the shape of the output.	keeps the data shape of the in-	
		put object. The input AO must	
		be an AO with row data and an	
		AO with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
00	Test the method with all data	Test that the power method	pass
09	objects.	works with cdata-, fsdata-,	
		tsdata-, and xydata objects	
		1) Check that the shpe of the	pass
		data doesn't change. 2) Check	
		that re-building of output is the	
		same as the output	

Table 100: Unit tests for a o/power.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/psd			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		1) Charle that wetterfor call and	
		1) Check that getinio call re-	pass
		turned an minio object in an	
		cases. 2) Check that all plass	
	Trata that the real mathed	The set of	
02	rests that the psd method	for a wasten of AOa againput	pass
	works with a vector of AOs as	1) Check that the number of al	pagg
	mput.	amonts in 'out' is the same as	pass
		in the input 2) Check that each	
		output object contains the cor-	
		rect values	
	Tests that the psd method	Test that the psd method works	pass
03	works with a matrix of AOs as	for a matrix of AOs as input.	Pass
	input.	1) Check that the number of el-	pass
	I and	ements in 'out' is the same as	1
		in the input. 2) Check that each	
		output object contains the cor-	
		rect values.	
0.4	Tests that the psd method	Test that the psd method works	pass
04	works with a list of AOs as	for a list of AOs as input.	
	input.	1) Check that the number of el-	pass
		ements in 'out' is the same as	
		in the input. 2) Check that each	
		output AO contains the correct	
		data.	
05	Tests that the psd method	Test that the psd method works	pass
	works with a mix of different	with an input of matrices and	
	shaped AOs as input.	vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
	Tests that the red method	AO contains the correct data.	
06	rests that the psd method	the red method can be precessed	pass
	property applies instory.	back to an m flo	
		1) Check that the last entry in	nagg
		the history of 'out' corresponds	pass
		to 'psd' 2) Check that the re-	
		built object is the same object as	
		'out'	
	Tests that the psd method	Test that the applying psd works	nass
07	agrees with MATLAB's nwelch	on a single AO.	Pass
	within some tolerance, when	1) Check that output agrees with	pass
	configured to use the same	the output of MATLAB's pwelch	
	parameters.	within tolerance.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/psd			
08	Check that the psd method pass	Call the method with a list of	pass
00	back the output objects to a list	output variables and with a sin-	
	of output variables or to a single	gle output variable. Additionaly	
	variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
09	Tests that the psd method	Test that the applying psd works	pass
	agrees with MATLAB's pwelch	on a single AO.	
	when configured to use the same	1) Check that output agrees	pass
	parameters.	with the output of MATLAB's	
		pwelch.	
10	Tests that the psd method	Test that the applying psd works	pass
	agrees with MATLAB's pwelch	on a single AO.	
	when configured to use the same	1) Check that output agrees	pass
	parameters.	with the output of MATLAB's	
		pweicn.	
11	Check that the [ao/psd] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		1) Check that the output.	D 0.00
		toing the same plotinfo plist	pass
	Tosts "conservation of operat":	1) Proparo the test tedata: white	neg
12	1) white poise produced from	noise from uniform distribution	pass
	uniform pdf with a given mean	\pm offset 2) Calculate the statis-	
	value and sigma (distribution's	tical parameters 3) Estimate the	
	1st and 2nd order) 2) evaluate	nsd	
	the sample mean value m and	1) Check that calculated psd en-	pass
	standard deviation s 3) psd of	ergy equals the rms content of	pass
	the white noise 4) compares the	the tsdata, estimated by the std	
	$\operatorname{sqrt}(\operatorname{sum}(S * df))$ with the	of the sample	
10	Standardingeriation of energy"	1) Prepare the test tsdata: white	pass
13	with fixed parameters: 1) white	noise from normal distribution +	-
	noise produced from normal	offset 2) Calculate the statistical	
	pdf, with a given mean value	parameters 3) Estimate the psd	
	and sigma (distribution's 1st	1) Check that calculated psd en-	pass
	and 2nd order) 2) evaluate the	ergy equals the rms content of	
	sample mean value m and	the tsdata, estimated by the std	
	standard deviation s 3) psd of	of the sample	
14	Testwlitenseiset4) nonfigueresythe	1) Prepare the test tsdata: white	pass
1 T.T	soutt(since(Spårdaf))etwish thewhite	noise from uniform distribution	
	stoiselgurd dieveid tioons uniform	+ offset 2) Calculate the statis-	
	pdf, with a given mean value	tical parameters 3) Estimate the	
	and sigma (distribution's 1st	psd	
	and 2nd order) 2) evaluate the		
	sample mean value m and		
	standard deviation s 3) psd of		
	the white noise 4) compares the		
	$\operatorname{sart}(\operatorname{sum}(S * df))$ with the		401

standard deviation s



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/psd			
		1) Check that calculated psd en-	pass
		ergy equals the rms content of	
		the tsdata, estimated by the std	
		of the sample	
15	Tests "conservation of energy"	1) Prepare the test tsdata: white	pass
10	with fixed parameters: 1) white	noise from normal distribution +	
	noise produced from normal	offset 2) Calculate the statistical	
	pdf, with a given mean value	parameters 3) Estimate the psd	
	and sigma (distribution's 1st	1) Check that calculated psd en-	pass
	and 2nd order) 2) evaluate the	ergy equals the rms content of	
	sample mean value m and	the tsdata, estimated by the std	
	standard deviation s 3) psd of	of the sample	
16	Tests litense is at 4 a compare sy the	1) Prepare the test tsdata: white	pass
10	south(suredSpårduh)etwish thewhite	noise from uniform distribution	
	staiselpud divid to ans uniform	+ offset 2) Calculate the statis-	
	pdf, with a given mean value	tical parameters 3) Estimate the	
	and sigma (distribution's 1st	psd	
	and 2nd order) 2) evaluate the	1) Check that calculated psd en-	pass
	sample mean value m and	ergy equals the rms content of	
	standard deviation s 3) psd of	the tsdata, estimated by the std	
	the white noise 4) compares the	of the sample	
17	Stepst(sum(Slingf)f) with:the white	1) Prepare the test tsdata: white	pass
11	staiselprodeleved from normal	noise from normal distribution +	
	pdf, with a given mean value	offset 2) Assign a random unit 3)	
	and sigma (distribution's 1st	PSD of the white noise	
	and 2nd orders) 2) PSD of the	1) Check that (calculated PSD	pass
	white noise 3) compares the	yunits) equals (input units) 2 /	
	units of the input and output	Hz 2) Check that (calculated	
		PSD xunits) equals Hz	
18	Tests handling of units: 1) white	1) Prepare the test tsdata: white	pass
	noise produced from uniform	noise from uniform distribution	
	pdf, with a given mean value	+ offset 2) Assign a random unit	
	and sigma (distribution's 1st	3) ASD of the white noise	
	and 2nd orders) 2) ASD of the	1) Check that (calculated ASD	pass
	white noise 3) compares the	yunits) equals (input units) /	
	units of the input and output	$H_{z}(1/2)$ 2) Check that (calcu-	
		lated ASD xunits) equals Hz	
19	Tests handling of units: 1) white	1) Prepare the test tsdata: white	pass
	noise produced from normal	noise from normal distribution +	
	pdf, with a given mean value	offset 2) Assign a random unit 3)	
	and sigma (distribution's 1st	PS of the white noise	
	and 2nd orders) 2) PS of the	1) Check that (calculated PS	pass
	white noise 3) compares the	yunits) equals (input units)2 2)	
	units of the input and output	Check that (calculated PS xu-	
		nits) equals Hz	
20	Lests handling of units: 1) white	1) Prepare the test tsdata: white	pass
	noise produced from uniform	noise from uniform distribution	
	distribution, with a given mean	+ offset 2) Assign a random unit	
	value and sigma (distribution's	3) AS of the white noise	
	1st and 2nd orders) 2) AS of the		
	white noise 3) compares the		
	units of the input and output		



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/psd			
		1) Check that (calculated AS	pass
		yunits) equals (input units) 2)	
		Check that (calculated AS xu-	
		nits) equals Hz	
91	Tests "conservation of energy":	1) Split the reference data into	pass
	1) white noise produced from	different segments 2) Calculates	
	normal pdf, with: a given mean	the PSD of the individual parts	
	value and sigma (distribution's	1) Evaluate the mean and stan-	pass
	1st and 2nd order) 2) Calculate	dard deviation of the Checks on	
	the expected level of noise from	individual PSDs: 2) Checks on	
	sample statistics 3) Calculates	individual PSDs: mean and stan-	
	the PSD of the individual parts,	dard deviation of the PSD points	
	with: Welch window and no	3) Evaluate the expected value,	
	detrend 4) Evaluate the mean	estimated from the std of the in-	
	and standard deviation of the	dividual segments 4) Compares	
	Checks on individual PSDs: 5)	with the mean performed on the	
	Checks on individual PSDs:	individual segments 5) Checks on	
	mean and standard deviation of	averaged PSDs: mean and stan-	
	the PSD points 6) Evaluate the	dard deviation of all the PSD	
	expected value, estimated from	points 6) Compares the grand-	
	the std of the individual	mean with the estimated white	
	segments 7) Compares with the	noise level	
22	Thests "penfoervadion of energy":	1) Split the reference data into	pass
	in)dividual sizenpentsus)dChecks	different segments 2) Calculates	
	oorawarqqetd WBDsangizeen and an	the PSD of the individual parts	
	standanddaiginti(nlisfräbhthon's	1) Evaluate the mean and stan-	pass
	PSDambi2ntsl 9)rComPaCaltheate	dard deviation of the Checks on	
	ghand-peeded wetlelt be not is material	individual PSDs: 2) Checks on	
	subinglenoisetiletices 3) Calculates	individual PSDs: mean and stan-	
	the PSD of the individual parts,	dard deviation of the PSD points	
	with: Welch window and mean	3) Evaluate the expected value,	
	detrend 4) Evaluate the mean	estimated from the std of the in-	
	and standard deviation of the	dividual segments 4) Compares	
	Checks on individual PSDs: 5)	with the mean performed on the	
	Checks on individual PSDs:	individual segments 5) Checks on	
	mean and standard deviation of	averaged PSDs: mean and stan-	
	the PSD points 6) Evaluate the	dard deviation of all the PSD	
	expected value, estimated from	points 6) Compares the grand-	
	the std of the individual	mean with the estimated white	
	segments 7) Compares with the	noise level	
23	inests recutervation thenergy":	1) Split the reference data into	pass
-	induviation of the industria and the industria a	different segments 2) Calculates	
	ooravaraged Whithsangeven and an	the PSD of the individual parts	
	standand degration of rabution's		
	Isstandizusi Srdeomparealcueate		
	stanuige en		
	the PSD of the individual parts,		
	with: Weich window and linear		
	detrend 4) Evaluate the mean		
	Checks on individual DCDs. ()		
	Checks on individual PSDs: 5)		
	Unecks on individual PSDs:		
	mean and standard deviation of the \mathbf{P} \mathbf{P} \mathbf{D} \mathbf{r} \mathbf{r} \mathbf{r}		493
	the F5D points 6) Evaluate the		
	expected value, estimated from		
	the sta of the individual		

segments 7) Compares with the



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/psd			
		1) Evaluate the mean and stan-	pass
		dard deviation of the Checks on	
		individual PSDs: 2) Checks on	
		individual PSDs: mean and stan-	
		dard deviation of the PSD points	
		3) Evaluate the expected value,	
		estimated from the std of the in-	
		dividual segments 4) Compares	
		with the mean performed on the	
		individual segments 5) Checks on	
		averaged PSDs: mean and stan-	
		dard deviation of all the PSD	
		points 6) Compares the grand-	
		mean with the estimated white	
		noise level	
	Tests "conservation of energy".	1) Split the reference data into	pass
24	1) white noise produced from	different segments 2) Calculates	pass
	normal pdf with a given mean	the PSD of the individual parts	
	value and sigma (distribution's	1) Evaluate the mean and stan-	pass
	1st and 2nd order) 2) Calculate	dard deviation of the Checks on	Pass
	the expected level of noise from	individual PSDs: 2) Checks on	
	sample statistics 3) Calculates	individual PSDs: p) encous en	
	the PSD of the individual parts.	dard deviation of the PSD points	
	with: Hanning window and no	3) Evaluate the expected value.	
	detrend 4) Evaluate the mean	estimated from the std of the in-	
	and standard deviation of the	dividual segments 4) Compares	
	Checks on individual PSDs: 5)	with the mean performed on the	
	Checks on individual PSDs:	individual segments 5) Checks on	
	mean and standard deviation of	averaged PSDs: mean and stan-	
	the PSD points 6) Evaluate the	dard deviation of all the PSD	
	expected value, estimated from	points 6) Compares the grand-	
	the std of the individual	mean with the estimated white	
	segments 7) Compares with the	noise level	
25	Thests performedion thenergy":	1) Split the reference data into	pass
20	in)divititieanloisgenparatsu&eCfreecks	different segments 2) Calculates	
	ooraværageti PADsangivæn andan	the PSD of the individual parts	
	standandddigiati(nlisfråbluthon's		
	RSDambi2ntsl 9) Compacealtheate		
	gnand-pæcaædwittelt be resisen atom		
	sample of set set is a set of the		
	the PSD of the individual parts,		
	with: Hanning window and		
	mean detrend 4) Evaluate the		
	mean and standard deviation of		
	the Checks on individual PSDs:		
	5) Checks on individual PSDs:		
	mean and standard deviation of $(b, c) = (b, c)$		
	the PSD points 6) Evaluate the		
	expected value, estimated from		
	the std of the individual		
	segments 7) Compares with the		
	individual sogments ?) Checks		
	on averaged PSDs: mean and		
	standard deviation of all the		494
	PSD points 9) Compares the		
	grand-mean with the estimated		
	white noise level		



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/psd			
		 Evaluate the mean and stan- dard deviation of the Checks on individual PSDs: 2) Checks on individual PSDs: mean and stan- dard deviation of the PSD points Evaluate the expected value, estimated from the std of the in- dividual segments 4) Compares with the mean performed on the individual segments 5) Checks on averaged PSDs: mean and stan- dard deviation of all the PSD points 6) Compares the grand- mean with the estimated white noise level 	pass
26	Tests "conservation of energy": 1) white noise produced from normal pdf, with: a given mean value and sigma (distribution's 1st and 2nd order) 2) Calculate the expected level of noise from sample statistics 3) Calculates the PSD of the individual parts, with: Hanning window and linear detrend 4) Evaluate the mean and standard deviation of the Checks on individual PSDs: 5) Checks on individual PSDs: mean and standard deviation of the PSD points 6) Evaluate the expected value, estimated from the std of the individual segments 7) Compares with the	 Split the reference data into different segments 2) Calculates the PSD of the individual parts Evaluate the mean and stan- dard deviation of the Checks on individual PSDs: 2) Checks on individual PSDs: mean and stan- dard deviation of the PSD points Evaluate the expected value, estimated from the std of the in- dividual segments 4) Compares with the mean performed on the individual segments 5) Checks on averaged PSDs: mean and stan- dard deviation of all the PSD points 6) Compares the grand- mean with the estimated white noise level 	pass
27	Tests jeculoenvadion ohenergy": h)divlidaaloisenpertsised Checks noraverageld PhDs angizen andan	1) Split the reference data into different segments 2) Calculates the PSD of the individual parts	pass
	standanddsiginationisfrähluthen's RSIAmpbi2ntsl 9) Compare altheate ghand-paceard withelthe restion at the schingelenstatikticsl 3) Calculates the PSD of the individual parts, with: Hamming window and no detrend 4) Evaluate the mean and standard deviation of the Checks on individual PSDs: 5) Checks on individual PSDs: mean and standard deviation of the PSD points 6) Evaluate the expected value, estimated from the std of the individual segments 7) Compares with the mean performed on the individual segments 8) Checks on averaged PSDs: mean and standard deviation of all the PSD points 9) Compares the grand-mean with the estimated white noise level		495



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/psd			
		1) Evaluate the mean and stan-	pass
		dard deviation of the Checks on	
		individual PSDs: 2) Checks on	
		individual PSDs: mean and stan-	
		dard deviation of the PSD points	
		3) Evaluate the expected value,	
		estimated from the std of the in-	
		dividual segments 4) Compares	
		with the mean performed on the	
		individual segments 5) Checks on	
		averaged PSDs: mean and stan-	
		dard deviation of all the PSD	
		points 6) Compares the grand-	
		mean with the estimated white	
		noise level	
	Tests "conservation of energy":	1) Split the reference data into	pass
28	1) white noise produced from	different segments 2) Calculates	-
	normal pdf, with: a given mean	the PSD of the individual parts	
	value and sigma (distribution's	1) Evaluate the mean and stan-	pass
	1st and 2nd order) 2) Calculate	dard deviation of the Checks on	-
	the expected level of noise from	individual PSDs: 2) Checks on	
	sample statistics 3) Calculates	individual PSDs: mean and stan-	
	the PSD of the individual parts,	dard deviation of the PSD points	
	with: Hamming window and	3) Evaluate the expected value,	
	mean detrend 4) Evaluate the	estimated from the std of the in-	
	mean and standard deviation of	dividual segments 4) Compares	
	the Checks on individual PSDs:	with the mean performed on the	
	5) Checks on individual PSDs:	individual segments 5) Checks on	
	mean and standard deviation of	averaged PSDs: mean and stan-	
	the PSD points 6) Evaluate the	dard deviation of all the PSD	
	expected value, estimated from	points 6) Compares the grand-	
	the std of the individual	mean with the estimated white	
	segments 7) Compares with the	noise level	
20	Thests reculoervadion thenergy":	1) Split the reference data into	pass
29	ib)divilitiean sizgnperotsu8d:Cfreeks	different segments 2) Calculates	
	noravat qgtti, PftDsangizan andan	the PSD of the individual parts	
	stand and digination is fraibuthen's		
	PSI Appbi2ntsI 9)r (Gov)n2)a (Galthe ate		
	ghanek-pæcked vietkelt bé nostion fatæd		
	wahitplenotsetiletives 3) Calculates		
	the PSD of the individual parts,		
	with: Hamming window and		
	linear detrend 4) Evaluate the		
	mean and standard deviation of		
	the Checks on individual PSDs:		
	5) Checks on individual PSDs:		
	mean and standard deviation of		
	the PSD points 6) Evaluate the		
	expected value, estimated from		
	the std of the individual		
	segments () Compares with the		
	mean performed on the		
	individual segments 8) Checks		
	on averaged PSDs: mean and		496
	Standard deviation of all the		
	roD points 9) Compares the		
	grand-mean with the estimated		
	white noise level		



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/psd			
		1) Evaluate the mean and stan- dard deviation of the Checks on individual PSDs: 2) Checks on individual PSDs: mean and stan- dard deviation of the PSD points 3) Evaluate the expected value, estimated from the std of the in- dividual segments 4) Compares with the mean performed on the individual segments 5) Checks on averaged PSDs: mean and stan- dard deviation of all the PSD points 6) Compares the grand- mean with the estimated white noise level	pass
30	Tests "conservation of energy": 1) white noise produced from normal pdf, with: a given mean value and sigma (distribution's 1st and 2nd order) 2) Calculate the expected level of noise from sample statistics 3) Calculates the PSD of the individual parts, with: BH92 window and no detrend 4) Evaluate the mean and standard deviation of the Checks on individual PSDs: 5) Checks on individual PSDs: mean and standard deviation of the PSD points 6) Evaluate the expected value, estimated from the std of the individual segments 7) Compares with the	 Split the reference data into different segments 2) Calculates the PSD of the individual parts Evaluate the mean and stan- dard deviation of the Checks on individual PSDs: 2) Checks on individual PSDs: mean and stan- dard deviation of the PSD points Evaluate the expected value, estimated from the std of the in- dividual segments 4) Compares with the mean performed on the individual segments 5) Checks on averaged PSDs: mean and stan- dard deviation of all the PSD points 6) Compares the grand- mean with the estimated white noise level 	pass
31	Thests "penfoervadion ohenergy": ih)divlideahoisenpartsu&dChecks noraværageti WHDs angiven andan	1) Split the reference data into different segments 2) Calculates the PSD of the individual parts	pass
	standandlæginti(nlisfråbhthen's PSDapoi2nts) 9) Comparealtheate ghand-paceaedwithelt of notion sahinplanoisatiktives 3) Calculates the PSD of the individual parts, with: BH92 window and mean detrend 4) Evaluate the mean and standard deviation of the Checks on individual PSDs: 5) Checks on individual PSDs: 5) Checks on individual PSDs: mean and standard deviation of the PSD points 6) Evaluate the expected value, estimated from the std of the individual segments 7) Compares with the mean performed on the individual segments 8) Checks on averaged PSDs: mean and standard deviation of all the PSD points 9) Compares the grand-mean with the estimated white noise level		497



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/psd			
		1) Evaluate the mean and stan-	pass
		individual DSDa 2) Charles on	
		individual PSDs: 2) Checks on	
		individual PSDs: mean and stan-	
		dard deviation of the PSD points	
		3) Evaluate the expected value,	
		estimated from the std of the in-	
		dividual segments 4) Compares	
		with the mean performed on the	
		individual segments 5) Checks on	
		averaged PSDs: mean and stan-	
		dard deviation of all the PSD	
		points 6) Compares the grand-	
		mean with the estimated white	
		noise level	
32	Tests "conservation of energy":	1) Split the reference data into	pass
02	1) white noise produced from	different segments 2) Calculates	
	normal pdf, with: a given mean	the PSD of the individual parts	
	value and sigma (distribution's	1) Evaluate the mean and stan-	pass
	1st and 2nd order) 2) Calculate	dard deviation of the Checks on	
	the expected level of noise from	individual PSDs: 2) Checks on	
	sample statistics 3) Calculates	individual PSDs: mean and stan-	
	the PSD of the individual parts,	dard deviation of the PSD points	
	with: BH92 window and linear	3) Evaluate the expected value,	
	detrend 4) Evaluate the mean	estimated from the std of the in-	
	and standard deviation of the	dividual segments 4) Compares	
	Checks on individual PSDs: 5)	with the mean performed on the	
	Checks on individual PSDs:	individual segments 5) Checks on	
	mean and standard deviation of	averaged PSDs: mean and stan-	
	the PSD points 6) Evaluate the	dard deviation of all the PSD	
	expected value, estimated from	points 6) Compares the grand-	
	the std of the individual	mean with the estimated white	
	segments () Compares with the	noise level	
33	the star premoenvention on energy :	different corrected 2) Coloristor	pass
	an and an and an and an and an and an and an	the PSD of the individual parts	
	stand and divisition in the stand and and divisition in the stand and divisition in the standard and and divisition in the standard and and and and and and and and and an	the FSD of the individual parts	
	BED maiored and another to the second state of		
	the manage man and which the left in the matrix is the section of		
	white a statil time a second		
	the PSD of the individual parts		
	with: Kaiser200 window and no		
	detrend 4) Evaluate the mean		
	and standard deviation of the		
	Checks on individual PSDs: 5)		
	Checks on individual PSDs:		
	mean and standard deviation of		
	the PSD points 6) Evaluate the		
	expected value, estimated from		
	the std of the individual		
	segments 7) Compares with the		
	mean performed on the		
	individual segments 8) Checks		
	on averaged PSDs: mean and		400
	standard deviation of all the		498
	PSD points 9) Compares the		
	grand-mean with the estimated		
	white noise level		



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/psd			
		1) Evaluate the mean and stan-	pass
		dard deviation of the Checks on	
		individual PSDs: 2) Checks on	
		individual PSDs: mean and stan-	
		dard deviation of the PSD points	
		3) Evaluate the expected value,	
		estimated from the std of the in-	
		dividual segments 4) Compares	
		with the mean performed on the	
		individual segments 5) Checks on	
		averaged PSDs: mean and stan-	
		dard deviation of all the PSD	
		points 6) Compares the grand-	
		mean with the estimated white	
		noise level	
34	1) molite using use level from	1) Split the reference data into	pass
	1) white hoise produced from	the DCD of the individual ports	
	when and sigma (distribution's	1) Evaluate the mean and stan	noga
	Value and Sigma (distribution's	dard deviation of the Checks on	pass
	the expected level of poice from	individual PSDs: 2) Checks on	
	sample statistics 3) Calculates	individual PSDs: 2) Checks on	
	the PSD of the individual parts	dard deviation of the PSD points	
	with: Kaiser200 window and	3) Evaluate the expected value	
	mean detrend 4) Evaluate the	estimated from the std of the in-	
	mean and standard deviation of	dividual segments 4) Compares	
	the Checks on individual PSDs:	with the mean performed on the	
	5) Checks on individual PSDs:	individual segments 5) Checks on	
	mean and standard deviation of	averaged PSDs: mean and stan-	
	the PSD points 6) Evaluate the	dard deviation of all the PSD	
	expected value, estimated from	points 6) Compares the grand-	
	the std of the individual	mean with the estimated white	
	segments 7) Compares with the	noise level	
25	Thests performation thenergy":	1) Split the reference data into	pass
00	ih)diwhicheanlosisgnperotsu&edCfreecks	different segments 2) Calculates	
	ooravatagett WADsangevan andan	the PSD of the individual parts	
	stand and diginti (nisfrabluthen's		
	PStlandi2ntsl DrdevinDatealtheate		
	gnanck-pecaed wettelt be restantation		
	sample of set set set 3) Calculates		
	the PSD of the individual parts,		
	linear detrend 4) Evaluate the		
	mean and standard deviation of		
	the Checks on individual PSDs:		
	5) Checks on individual PSDs:		
	mean and standard deviation of		
	the PSD points 6) Evaluate the		
	expected value, estimated from		
	the std of the individual		
	segments 7) Compares with the		
	mean performed on the		
	individual segments 8) Checks		
	on averaged PSDs: mean and		400
	standard deviation of all the		499
	PSD points 9) Compares the		
	grand-mean with the estimated		
	white noise level		



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/psd			
		1) Evaluate the mean and stan-	pass
		dard deviation of the Checks on	
		individual PSDs: 2) Checks on	
		individual PSDs: mean and stan-	
		dard deviation of the PSD points	
		3) Evaluate the expected value,	
		estimated from the std of the in-	
		dividual segments 4) Compares	
		with the mean performed on the	
		individual segments 5) Checks on	
		averaged PSDs: mean and stan-	
		dard deviation of all the PSD	
		points 6) Compares the grand-	
		mean with the estimated white	
		noise level	
28	Tests "conservation of energy":	1) Prepare the test tsdata: white	pass
30	1) white noise produced from	noise from normal distribution +	
	normal pdf, with: a given mean	offset 2) Calculate the statistical	
	value and sigma (distribution's	parameters 3) Add a trend 4) Es-	
	1st and 2nd order) 2) evaluate	timate the psd	
	the sample mean value m and	1) Check that calculated psd en-	pass
	standard deviation s 3) add a	ergy equals the rms content of	
	given trend of order n 4) psd of	the tsdata, estimated by the std	
	the noise, with proper	of the sample	
39	destendingservationpatientergy":	1) Prepare the test tsdata: white	pass
	sort (site to ised b) out that be	noise from uniform distribution	
	straubotandpotey wittom as given mean	+ offset 2) Calculate the statis-	
	value and sigma (distribution's	tical parameters 3) Add a trend	
	1st and 2nd order) 2) evaluate	4) Estimate the psd	
	the sample mean value m and	1) Check that calculated psd en-	pass
	standard deviation s 3) add a	the tadata estimated by the atd	
	the poise with proper	of the sample	
	The noise, with proper	1) Propage the test tedate: white	neg
40	desistation (Sized Stratic bet here)	noise from normal distribution +	pass
	standard fewist tons given mean	offset 2) Calculate the statistical	
	value and sigma (distribution's	parameters 3) Add a trend 4) Es-	
	1st and 2nd order) 2) evaluate	timate the psd	
	the sample mean value m and	1) Check that calculated psd en-	pass
	standard deviation s 3) add a	ergy equals the rms content of	Pass
	given trend of order n 4) psd of	the tsdata, estimated by the std	
	the noise, with proper	of the sample	
11	Testendie effect of pairesothieng: 1)	1) Prepare the test tsdata: white	pass
41	soprit (sumiss profil) within the	noise from normal distribution +	-
	standarødfewiationasgiven mean	offset 2) Calculate the statistical	
	value and sigma (distribution's	parameters 3) Estimate the psd	
	1st and 2nd order) 2) psd of the	without detrending, Rectangular	
	noise, without detrending,	window 4) Manually apply win-	
	Rectangular window 3) Apply	dow to the data 5) Estimate the	
	the detrending and the window	psd without detrending, Rectan-	
	manually 4) psd of the noise,	gular window	
	without detrending, Rectangular		
	window 5) compares the to psds		



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/psd			
		1) Check that calculated psds are	pass
		identical within a part in $10\hat{1}2$	
49	Tests the effect of windowing: 1)	1) Prepare the test tsdata: white	pass
42	white noise produced from	noise from normal distribution +	
	normal pdf, with: a given mean	offset 2) Calculate the statistical	
	value and sigma (distribution's	parameters 3) Estimate the psd	
	1st and 2nd order) 2) psd of the	without detrending, BH92 win-	
	noise, without detrending, BH92	dow 4) Manually apply window	
	window 3) Apply the detrending	to the data 5) Estimate the psd	
	and the window manually 4)	without detrending, Rectangular	
	psd of the noise, without	window	
	detrending, Rectangular window	1) Check that calculated psds are	pass
	5) compares the to psds	identical within a part in $10\hat{1}2$	-
40	Tests the effect of windowing: 1)	1) Prepare the test tsdata: white	pass
43	white noise produced from	noise from normal distribution +	-
	normal pdf, with: a given mean	offset 2) Calculate the statisti-	
	value and sigma (distribution's	cal parameters 3) Estimate the	
	1st and 2nd order) 2) psd of the	psd without detrending, Ham-	
	noise, without detrending,	ming window 4) Manually ap-	
	Hamming window 3) Apply the	ply window to the data 5) Es-	
	detrending and the window	timate the psd without detrend-	
	manually 4) psd of the noise,	ing, Rectangular window	
	without detrending, Rectangular	1) Check that calculated psds are	pass
	window 5) compares the to psds	identical within a part in $10\hat{1}2$	-
4.4	Tests the effect of windowing: 1)	1) Prepare the test tsdata: white	pass
44	white noise produced from	noise from normal distribution +	
	normal pdf, with: a given mean	offset 2) Calculate the statisti-	
	value and sigma (distribution's	cal parameters 3) Estimate the	
	1st and 2nd order) 2) psd of the	psd without detrending, Hanning	
	noise, without detrending,	window 4) Manually apply win-	
	Hanning window 3) Apply the	dow to the data 5) Estimate the	
	detrending and the window	psd without detrending, Rectan-	
	manually 4) psd of the noise,	gular window	
	without detrending, Rectangular	1) Check that calculated psds are	pass
	window 5) compares the to psds	identical within a part in $10\hat{1}2$	
45	Tests the effect of windowing: 1)	1) Prepare the test tsdata: white	pass
40	white noise produced from	noise from normal distribution +	
	normal pdf, with: a given mean	offset 2) Calculate the statisti-	
	value and sigma (distribution's	cal parameters 3) Estimate the	
	1st and 2nd order) 2) psd of the	psd without detrending, Bartlett	
	noise, without detrending,	window 4) Manually apply win-	
	Bartlett window 3) Apply the	dow to the data 5) Estimate the	
	detrending and the window	psd without detrending, Rectan-	
	manually 4) psd of the noise,	gular window	
	without detrending, Rectangular	1) Check that calculated psds are	pass
	window 5) compares the to psds	identical within a part in $10\hat{1}2$	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/psd			
46	Tests the effect of windowing: 1)	1) Prepare the test tsdata: white	pass
40	white noise produced from	noise from normal distribution +	
	normal pdf, with: a given mean	offset 2) Calculate the statisti-	
	value and sigma (distribution's	cal parameters 3) Estimate the	
	1st and 2nd order) 2) psd of the	psd without detrending, Nuttall3	
	noise, without detrending,	window 4) Manually apply win-	
	Nuttall3 window 3) Apply the	dow to the data 5) Estimate the	
	detrending and the window	psd without detrending, Rectan-	
	manually 4) psd of the noise,	gular window	
	without detrending, Rectangular	1) Check that calculated psds are	pass
	window 5) compares the to psds	identical within a part in $10\hat{1}2$	
47	Tests the effect of windowing: 1)	1) Prepare the test tsdata: white	pass
41	white noise produced from	noise from normal distribution +	
	normal pdf, with: a given mean	offset 2) Calculate the statistical	
	value and sigma (distribution's	parameters 3) Estimate the psd	
	1st and 2nd order) 2) psd of the	without detrending, Kaiser psll	
	noise, without detrending,	= random window 4) Manually	
	Kaiser $psll = random window 3$)	apply window to the data 5) Es-	
	Apply the detrending and the	timate the psd without detrend-	
	window manually 4) psd of the	ing, Rectangular window	
	noise, without detrending,	1) Check that calculated psds are	pass
	Rectangular window 5)	identical within a part in $10\hat{1}2$	
18	Textpates effectoopsdindowing: 1)	1) Prepare the test tsdata: white	pass
40	white noise produced from	noise from normal distribution +	
	normal pdf, with: a given mean	offset 2) Calculate the statistical	
	value and sigma (distribution's	parameters 3) Estimate the psd	
	1st and 2nd order) 2) psd of the	without detrending, Kaiser psll	
	noise, without detrending,	= default window 4) Manually	
	Kaiser $psll = default window 3$)	apply window to the data 5) Es-	
	Apply the detrending and the	timate the psd without detrend-	
	window manually 4) psd of the	ing, Rectangular window	
	noise, without detrending,	1) Check that calculated psds are	pass
	Rectangular window 5)	identical within a part in $10\hat{1}2$	
49	Tourstpathes effectoopsdandowing: 1)	1) Prepare the test tsdata: white	pass
	white noise produced from	noise from normal distribution +	
	normal pdf, with: a given mean	offset 2) Calculate the statisti-	
	value and sigma (distribution's	cal parameters 3) Estimate the	
	1st and 2nd order) 2) psd of the	psd without detrending, Nuttall4	
	noise, without detrending,	window 4) Manually apply win-	
	Nuttall4 window 3) Apply the	dow to the data 5) Estimate the	
	detrending and the window	psd without detrending, Rectan-	
	manually 4) psd of the noise,	gular window	
	without detrending, Rectangular	1) Check that calculated psds are	pass
	window 5) compares the to psds	identical within a part in $10\hat{1}2$	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/psd			
50	Tests the effect of windowing: 1)	1) Prepare the test tsdata: white	pass
	white noise produced from	noise from normal distribution +	
	normal pdf, with: a given mean	offset 2) Calculate the statistical	
	value and sigma (distribution's	parameters 3) Estimate the psd	
	1st and 2nd order) 2) psd of the	without detrending, SFT3F win-	
	noise, without detrending,	dow 4) Manually apply window	
	SFT3F window 3) Apply the	to the data 5) Estimate the psd	
	detrending and the window	without detrending, Rectangular	
	manually 4) psd of the noise,	window	
	without detrending, Rectangular	1) Check that calculated psds are	pass
	window 5) compares the to psds	identical within a part in 1012	
51	Tests the possibility to set the	1) Prepare the test tsdata: white	pass
01	number of averages rather than	noise from normal distribution +	
	setting the Nfft: 1) white noise	offset 2) psd of the noise, without	
	produced from normal pdf,	detrending, random window, set	
	with: a given mean value and	number of averages	
	sigma (distribution's 1st and	1) Check that calculated navs are	pass
	2nd order) 2) psd of the noise,	identical to those requested	
52	Westsoutheletossibility tandomhe	1) white noise produced from	pass
	window, ofet verages refleverages	normal pdf, with: a given mean	
	SettingcklikeNeffect)vehitenbeisef	value and sigma (distribution's	
	proclages from normal pdf,	1st and 2nd order) 2) psd of the	
	with: a given mean value and	noise, without detrending, ran-	
	sigma (distribution's 1st and	dom window, random navs 3) get	
	2nd order) 2) psd of the noise,	the number of averages 4) get the	
	without detrending, random	nfft used 5) run psd again, with	
	window, random navs 3) get the	the nfit used	
	number of averages 4) get the	1) Uneck that calculated objects	pass
	The traff wash bit it again, with	1) white poise produced from	naga
53	and the states of the second s	normal pdf with: a given mean	pass
	sotting the Nfft: 1) white poise	value and sigma (distribution's	
	produced from normal pdf	1st and 2nd order) 2) psd of the	
	with a given mean value and	noise without detrending ran-	
	sigma (distribution's 1st and	dom window random navs 3) get	
	2nd order) 2) psd of the poise	the number of averages 4) get	
	without detrending random	the nfft used 5) run psd again	
	window random navs 3) get the	with the nfft used 6) run psd	
	number of averages 4) get the	again, with conflicting parame-	
	nfft used 5) run psd again, with	ters, and verify it uses nfft rather	
	the nfft used 6) compare navs.	than navs	
	nfft, psds	1) Check that calculated objects	pass
		S1 and S2 are identical 2) Check	1
		that S3 used different values	
F 4	Tests "conservation of energy":	1) Prepare the test tsdata: white	pass
04	1) white noise produced from	noise from normal distribution +	-
	normal pdf, with a given mean	offset 2) Calculate the statistical	
	value and sigma (distribution's	parameters 3) Estimate the psd	
	1st and 2nd order) 2) evaluate		
	the sample mean value m and		
	standard deviation s 3) psd of		
	the white noise 4) compares the		
	$\operatorname{sqrt}(\operatorname{sum}(S * df))$ with the		
	standard deviation s		502



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/psd			
		1) Check that calculated psd en-	pass
		ergy equals the rms content of	
		the tsdata, estimated by the std	
		of the sample	
60	Tests the 'basic' call: 1) white	1) Prepare the test tsdata: white	pass
00	noise produced from normal	noise from normal distribution +	
	pdf, with: a given mean value	offset 2) Calculate the statistical	
	and sigma (distribution's 1st	parameters 3) Estimate the psd	
	and 2nd order) 2) psd of the	with default detrending, default	
	noise, defualt detrending,	window, no averaging	
	default window, no averaging	1) Nothing to check	pass
61	Tests the 'basic' call: 1) white	1) Prepare the test tsdata: white	pass
01	noise produced from normal	noise from normal distribution +	
	pdf, with: a given mean value	offset 2) Calculate the statistical	
	and sigma (distribution's 1st	parameters 3) Estimate the psd	
	and 2nd order) 2) psd of the	with default detrending, default	
	noise, defualt detrending,	window, no averaging	
	default window, no averaging	1) Nothing to check	pass
62	Tests the 'basic' call: 1) white	1) Prepare the test tsdata: white	pass
02	noise produced from normal	noise from normal distribution +	
	pdf, with: a given mean value	offset 2) Calculate the statistical	
	and sigma (distribution's 1st	parameters 3) Estimate the psd	
	and 2nd order) 2) psd of the	with default detrending, default	
	noise, defualt detrending,	window, no averaging	
	default window, no averaging	1) Nothing to check	pass
63	Tests the 'basic' call: 1) white	1) Prepare the test tsdata: white	pass
00	noise produced from normal	noise from normal distribution +	
	pdf, with: a given mean value	offset 2) Calculate the statistical	
	and sigma (distribution's 1st	parameters 3) Estimate the psd	
	and 2nd order) 2) psd of the	with default detrending, default	
	noise, defualt detrending,	window, no averaging	
	default window, no averaging	1) Nothing to check	pass

Table 101: Unit tests for ao/psd.


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/rdivide			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
rule1 (tsdata and tsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and fsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule1 (cdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (tsdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (cdata and tsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule1 (xydata and	Tests the arithmetic operators	Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
tsdata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule1 (cdata and	Tests the arithmetic operators	Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
fsdata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	Pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and fsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule1 (cdata and	Tests the arithmetic operators	Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
xydata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single tsdata and vector tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector tsdata and single tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule2 (cingle	Tasts the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata and vector tsdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector tsdata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single cdata and vector tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule2 (vector tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and single cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single fsdata and vector fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector fsdata and single fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single xydata and vector fsdata)	rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector fsdata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single cdata and vector fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule2 (vector fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and single cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single xydata and vector xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector xydata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule2 (single cdate	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector xydata and single cdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector cdata and single cdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule2 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rulo3 (vector tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	Pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule3 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector tsdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	разы
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule3 (vector fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector xydata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule3 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector fsdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	1
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule3 (vector	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata and vector cdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector cdata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector cdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule3 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	1
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule4 (vector tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector xydata)	rule 4.	rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	Ĩ
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule4 (vector fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector fsdata)	rule 4.	rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	Ĩ
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector fsdata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector fsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector xydata and vector xydata)	rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector xydata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule4 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 4.	rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP tsdata and single tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single tsdata and NxP tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule5 (NxP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and single xydata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	1
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP tsdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule5 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP tsdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP fsdata and single fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single fsdata and NxP fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule5 (NxP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and single xydata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	1
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP fsdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rulo5 (single edata	Tasts the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP xydata and single xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule5 (NxP xydata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and single cdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single cdata and NxP xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP cdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule5 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP cdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule6 (NxP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	P
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule6 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP tsdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule6 (NxP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rulo6 (vostor adata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rulo6 (NyP yydata	Tosts the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector cdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule6 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP cdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule7 (NxP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector xydata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rulo7 (voctor edata	Tosts the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule7 (NxP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector xydata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	-
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule7 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP fsdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule7 (NxP xvdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	Ĩ
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule7 (voeter adote	Tosts the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP cdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule8 (NxP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector xydata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	1
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rulos (vostor adota	Tasts the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule8 (NxP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector xydata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
mula? (vestor state	Tests the arithmetic encretors	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule8 (NxP xvdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	Ĩ
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule8 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP cdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	P
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP tsdata and NxQ tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ tsdata and NxP tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule9 (NxP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxQ xydata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	1
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP tsdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
mile0 (NrO edete	Tests the arithmetic encretors	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP fsdata and NxQ fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ fsdata and NxP fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
mule0 (NuP fedete	Tasts the exithmetic encretors	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxQ xydata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP fsdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule9 (NxQ cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP fsdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	1
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP xydata and NxQ xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
wild (NyP widete	Tasts the exithmetic encreters	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxQ cdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ cdata and NxP xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP cdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
wile0 (NirO edete	Tosts the arithmetic encreters	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP cdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
mile10 (NuD todate	Tasts the exithmatic encretors	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP xydata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rula10 (NyP adata	Tests the arithmetic operators	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/rdivide			
rule10 (NxP fsdata and NxP xvdata)	Tests the arithmetic operators rule 10.	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators rule 10 for each data type; xy-	pass
		data, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rula10 (NyP adota	Tasts the arithmetic operators	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide			
rule10 (NyP	Tests the arithmetic operators	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata and NxP cdata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/rdivide			
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rdivide			pass
tosta (fadata and	Tests all arithmatic an anations	Tests all arithmatic energians	pass
tsdata)	which are not allowed.	which are not allowed.	pass
		Here we test some negative cases.	pass
tests (tsdata and fsdata)	Tests all arithmetic operations which are not allowed.	Tests all arithmetic operations which are not allowed.	pass
(1.2.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1		Here we test some negative cases.	pass
tests (AO no data and tsdata)	Tests all arithmetic operations which are not allowed.	Tests all arithmetic operations which are not allowed.	pass
		Here we test some negative cases.	pass
AO no data)	which are not allowed.	which are not allowed.	pass
		Here we test some negative cases.	pass
tsdata)	which are not allowed.	which are not allowed.	pass
		Here we test some negative cases.	pass
tests (different x	Tests all arithmetic operations	Tests all arithmetic operations	pass
values in fsdata)	which are not allowed.	which are not allowed. Here we test some negative cases.	pass
tests (negative	Tests all arithmetic operations	Tests all arithmetic operations	pass
test)	which are not allowed.	which are not allowed.	
		Here we test some negative cases.	pass
tests (negative	1ests all arithmetic operations	Tests all arithmetic operations	pass
test)	which are not allowed.	which are not allowed.	
		Here we test some negative cases.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/rdivide

Table 102: Unit tests for a o/rdivide.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/real			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters	pass
02	Tests that the real method works with a vector of AOs as	Test that the real method works for a vector of AOs as input.	pass
	input.	1) Check that the number of el- ements in 'out' is the same as in 'atvec' 2) Check that each output AO contains the correct data.	pass
03	Tests that the real method works with a matrix of AOs as input	Test that the real method works for a matrix of AOs as input.	pass
	mput.	ements in 'out' is the same as in 'atmat' 2) Check that each output AO contains the correct data.	pass
04	Tests that the real method works with a list of AOs as input	Test that the real method works for a list of AOs as input.	pass
	mput.	ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
05	Tests that the real method works with a mix of different shaped AOs as input.	Test that the real method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the real method properly applies history.	Test that the result of applying the real method can be processed back to an m-file.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'real'. 2) Check that the re- built object is the same object as 'out'.	pass
07	Tests that the real method can modify the input AO.	Test that the real method can modify the input AO by call- ing with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass
		1) Check that 'at1' and 'ain' are now different. 2) Check that 'ain' is real(at1).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/real			1
08	Control the method with a plict	Test that the real method can	pass
00	Control the method with a plist.	modify the single axis controlled	
		by the plist and the resuld can be	
		processed back to an m-file.	
		1) Check that the real method	pass
		applies to the x-axis 2) Check	
		that the real method applies to	
		the y-axis 3) Check that the real	
		method applies to both axes 4)	
		Check that the re-built objects	
		are the same object as ' $out[13]$ '.	
00	Control the method with a plict	Test that the real method keeps	pass
09	Control the method with a plist.	the data shape of the input ob-	
		ject. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
10	Check that the real method pass	Call the method with a list of	pass
10	back the output objects to a list	output variables and with a sin-	
	of output variables or to a single	gle output variable. Additionaly	
	variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/real] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 103: Unit tests for ao/real.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/rebuild			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the rebuild method	Test that the rebuild method	pass
02	works with a vector of AO	works for a vector of AO objects	
	objects as input.	as input.	
		1) Check the rebuilt output.	pass
0.2	Tests that the rebuild method	Test that the rebuild method	pass
03	works with a matrix of AO	works for a matrix of AO objects	
	objects as input.	as input.	
		1) Check the rebuilt output.	pass
0.4	Tests that the rebuild method	Test that the rebuild method	pass
04	works with a list of AO objects	works for a list of AO objects as	-
	as input.	input.	
	_	1) Check the rebuilt output.	pass
05	Tests that the rebuild method	Test that the rebuild method	pass
05	works with a mix of different	works with an input of matrices	-
	shaped AO objects as input.	and vectors and single AO ob-	
		jects.	
		1) Check the rebuilt output.	pass
0.0	Tests that the rebuild method	The method rebuild doesn't	pass
06	properly applies history.	change the data, thus it is not	1
		possible to check the history.	
		Nothing to do.	
			pass
07	Check that the rebuild method	Call the method with a list of	pass
07	pass back the output objects to	output variables and with a sin-	-
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/rebuild]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	-

Table 104: Unit tests for ao/rebuild.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/resample			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.9	Tests that the resample method	Test that the resample method	pass
02	works with a vector of AOs as	works for a vector of AOs as in-	
	input.	put.	
		1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
02	Tests that the resample method	Test that the resample method	pass
05	works with a matrix of AOs as	works for a matrix of AOs as in-	
	input.	put.	
		1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
0.4	Tests that the resample method	Test that the resample method	pass
04	works with a list of AOs as	works for a list of AOs as input.	
	input.	1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
05	Tests that the resample method	Test that the resample method	pass
05	works with a mix of different	works with an input of matrices	
	shaped AOs as input.	and vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
06	Tests that the resample method	Test that the result of applying	pass
	properly applies history.	the resample method can be pro-	
		cessed back to an m-file.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'resample'. 2) Check that the	
		re-built object is the same object	
		as 'out'.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/resample			
07	Tests that the resample method	Test that the resample method	pass
	can modify the input AO.	can modify the input AO by call-	
		ing with no output and that the	
		method doesn't change the input	
		of the function notation (with a	
		equal sign).	
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		is resample(at1).	
08	Control the method with a plict	Test that the resample method	pass
08	Control the method with a plist.	keeps the data shape of the in-	
		put object. The input AO must	
		be an AO with row data and an	
		AO with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
00	Check that the resample	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/resample]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 105: Unit tests for ao/resample.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/rms			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.9	Tests that the rms method	Test that the rms method works	pass
02	works with a vector of AOs as	for a vector of AOs as input.	
	input.	1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
	Tests that the rms method	Test that the rms method works	pass
03	works with a matrix of AOs as	for a matrix of AOs as input.	P
	input.	1) Check that the number of ele-	pass
	F	ments in 'out' is the square of the	P
		number in the input. 2) Check	
		that each output AO contains	
		the correct data	
	Tests that the rms method	Test that the rms method works	nass
04	works with a list of $\Delta \Omega_{\rm S}$ as	for a list of $\Delta \Omega_{\rm S}$ as input	pass
	input	1) Check that the number of ele	nase
	mput.	monta in 'out' is the square of the	pass
		number in the input 2) Check	
		that each autput AO contains	
		that each output AO contains	
	Tests that the wms method	The correct data.	magg
05	results unit the rms method	Test that the rms method works	pass
	works with a mix of different	with an input of matrices and	
	snaped AOs as input.	vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
06	Tests that the rms method	Test that the result of applying	pass
	properly applies history.	the rms method can be processed	
		back to an m-file.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'rms'. 2) Check that the re-	
		built object is the same object as	
		'out'.	
07	Tests that the rms method can	Test that the rms method can	pass
	modify the input AO.	modify the input AO by call-	
		ing with no output and that the	
		method doesn't change the input	
		of the function notation (with a	
		equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/rms			
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		is rms(at1). 3) Check the algo-	
		rithm	
00	Control the method with a plict	Test that the rms method keeps	pass
08	Control the method with a plist.	the data shape of the input ob-	
		ject. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	-
00	Check that the rms method pass	Call the method with a list of	pass
09	back the output objects to a list	output variables and with a sin-	-
	of output variables or to a single	gle output variable. Additionaly	
	variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	-
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/rms] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	1

Table 106: Unit tests for a o/rms.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/round			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the [ao/round]	Test that the [ao/round] method	pass
	method works with a vector of	works for a vector of objects as	
	objects as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		vec' 2) Check that each output	
		object contains the correct data.	
03	Tests that the [ao/round]	Test that the [ao/round] method	pass
	method works with a matrix of	works for a matrix of objects as	
	objects as input.	1) Cl 1 + 1 + 1	
		1) Check that the number of el-	pass
		ments in out is the same as in	
		abject contains the correct data	
	Tests that the [as /round]	Test that the loo /round] method	paga
04	method works with a list of	works for a list of abjects as in	pass
	objects as input	works for a list of objects as in-	
	objects as input.	1) Check that the number of el-	nase
		ements in 'out' is the same as in	pass
		'mat' 2) Check that each output	
		object contains the correct data	
	Tests that the [ao/round]	Tests that the [ao/round]	pass
05	method works with a mix of	method works with a mix of	pass
	different arrays of objects as	different arrays of objects as	
	input.	input.	
	I m	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
06	Tests that the [ao/round]	Test that the result of applying	pass
00	method properly applies history.	the [ao/round] method can be	
		processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[ao/round]'. 2) Check that the	
		re-built object is the same object	
		as the input.	
07	Tests that the [ao/round]	Test that the [ao/round] method	pass
	method can modify the input	can modify the input object by	
	AO.	calling with no output and that	
		the method doesn't change the	
		input of the function notation	
		(with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/round			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the [ao/round]	
		value of the copy 4) Check that	
		out and amodi are the same	
	Test that the [ao/round] method	Test that the [ao/round] method	pass
08	uses the plist to get the axis.	uses the plist to get the axis.	-
		1) Check that the [ao/round]	pass
		method applies to the x-axis	1
		2) Check that the [ao/round]	
		method applies to the y-axis	
		3) Check that the [ao/round]	
		method applies to both axes 4)	
		Check that the re-built object is	
		the same as in ' $out[13]$ '.	
00	Test the shape of the data in	Test that the [ao/round] method	pass
09	AOs.	keeps the data shape of the input	-
		object. The input AO data must	
		be an array with row data and/or	
		column data.	
		1) Check that the shape of the	pass
		data doesn't change.	-
10	Check that the [ao/round]	Call the method with a list of	pass
10	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/round]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 107: Unit tests for a o/round.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/sDomainFit			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re-	pass
		turned an minfo object in all	Pass
		cases 2) Check that all plists	
		have the correct parameters	
	Tosts that the sDomainFit	Tost that the sDomainFit	nage
02	method works with a vostor of	method works for a vostor of	pass
	A Og ag input	And a simplet	
	AOS as input.	1) Check that the number of al	naga
		amonta in 'out' is the same as in	pass
		'and 2) Check that each autout	
		AQ contained the compatible	
		AO contains the correct data.	
03	lests that the sDomainFit	lest that the sDomainFit	pass
	method works with a matrix of	method works for a matrix of	
	AOs as input.	AOs as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		(am' 2) Check that each output	
		AO contains the correct data.	
04	Tests that the sDomainFit	Test that the sDomainFit	pass
	method works with a list of AOs	method works for a list of AOs	
	as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
05	Tests that the sDomainFit	Test that the sDomainFit	pass
00	method works with a mix of	method works with an input of	
	different shaped AOs as input.	matrices and vectors and single	
		AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
06	Tests that the sDomainFit	Test that the result of applying	pass
	method properly applies history.	the sDomainFit method can be	
		processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'sDomainFit'. 2) Check that	
		the re-built object is the same	
		object as the input.	
07	sDomainFit cannot modify the	Test that the sDomainFit	pass
07	input AO.	method can modify the input	-
	-	AO by calling with no output	
		and that the method doesn't	
		change the input of the function	
		notation (with a equal sign).	
1		(



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/sDomainFit			
,		1) Nothing to do.	pass
09	Check that the sDomainFit method pass back the output objects to a single variable	Call the method with a list of output variables and with a sin- gle output variable. Additionaly	pass
	correctly.	check that the rebuild method works on the output.	
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild'	pass
		method produces the same object as 'out'.	
10	Tests that the sDomainFit method return the correct coefficients	Test that the sDomainFit method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that output con- tains the correct coefficients.	pass

Table 108: Unit tests for ao/sDomainFit.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/save			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.0	Tests that the save method	Test that the save method works	pass
02	works with a vector of AOs as	for a vector of AOs as input. Test	
	input.	both formats 'xml' and 'mat'.	
		1) Check that the number of	pass
		elements in 'out1' and 'out2'	-
		are the same as in 'atvec' 2)	
		Check that the loaded objects	
		are the same as the saved ob-	
		jects except the the history be-	
		cause the load-constructor adds	
		one history step. 3) The outputs	
		'out1' and 'out2' must be the	
		same except the history proper-	
		ties 'methodInfo'. 'plistUsed' and	
		'proctime'.	
0.0	Tests that the save method	Test that the save method works	pass
03	works with a matrix of AOs as	for a matrix of AOs as input.	1
	input.	Test both formats 'xml' and	
	-	'mat'.	
		1) Check that the number of	pass
		elements in 'out1' and 'out2'	1
		are the same as in 'atmat' 2)	
		Check that the loaded objects	
		are the same as the saved ob-	
		jects except the the history be-	
		cause the load-constructor adds	
		one history step. 3) The outputs	
		'out1' and 'out2' must be the	
		same except the history proper-	
		ties 'methodInfo', 'plistUsed' and	
		'proctime'.	
	Tests that the save method	Test that the save method works	pass
04	works with a list of AOs as	for a list of AOs as input. Test	T
	input.	both formats 'xml' and 'mat'.	
	I		



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/save			
		1) Check that the number of elements in 'out1' and 'out2' are the same as in the list 2) Check that the loaded objects are the same as the saved ob- jects except the the history be- cause the load-constructor adds one history step. 3) The outputs 'out1' and 'out2' must be the same except the history proper- ties 'methodInfo', 'plistUsed' and 'proctime'.	pass
05	Tests that the save method works with a mix of different shaped AOs as input.	Test that the save method works with an input of matrices and vectors and single AOs. Test both formats 'xml' and 'mat'. 1) Check that the number of el- ements in 'out' is the same as in	pass
		AO contains the correct data.	
06	Tests that the save method properly applies history.	Test that the result of apply- ing the save method can be pro- cessed back to an m-file. Do this for both extensions 'mat' and 'yml'	pass
		1) Check that save doesn't add a history step. 2) Check that the read object is the same as the stored object. save + load doesn't add history. 3) Check that the re-built object is the same object as 'out'.	pass
07	Tests that the save method works with the modify command.	Test that the modify command. 1) Check that the save method applies the history. 2) Check the output against the input except the history. 3) Check the history of the output against the input.	pass
08	Control the method with a plist.	Test that the save method uses the filename which is stored in a plist.	pass
09	Test the save method with an AO with complex data.	Save an AO with complex fsdata object. 1) Check the output	pass
10	Test the save method with an AO which is created from a pole/zero model	Save an AO which is created from a pzmodel.	pass
11	Test the save method with a complex plist which builds the	Save an AO which is created from a complex plist.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/save		
	1) Check the output	pass

Table 109: Unit tests for ao/save.


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/scale			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the [ao/scale] method works with a vector of objects as input.	Test that the [ao/scale] method works for a vector of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'vec' 2) Check that each output object contains the correct data.	pass
03	Tests that the [ao/scale] method works with a matrix of objects as input.	Test that the [ao/scale] method works for a matrix of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
04	Tests that the [ao/scale] method works with a list of objects as input.	Test that the [ao/scale] method works for a list of objects as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
05	Tests that the [ao/scale] method works with a mix of different arrays of objects as input.	Tests that the [ao/scale] method works with a mix of different ar- rays of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
06	Tests that the [ao/scale] method properly applies history.	Test that the result of applying the [ao/scale] method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[ao/scale]'. 2) Check that the re-built object is the same object as the input.	pass
07	Tests that the [ao/scale] method can modify the input AO.	Test that the [ao/scale] method can modify the input object by calling with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/scale			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the [ao/scale]	
		value of the copy 4) Check that	
		out and amodi are the same	
00	Test the shape of the data in	Test that the [ao/scale] method	pass
09	AOs.	keeps the data shape of the input	
		object. The input AO data must	
		be an array with row data and/or	
		column data.	
		1) Check that the shape of the	pass
		data doesn't change.	
10	Check that the [ao/scale]	Call the method with a list of	pass
10	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/scale]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
	Ĭ	works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 110: Unit tests for ao/scale.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/search			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.0	Tests that the search method	Test that the search method	pass
02	works with a vector of AOs as	works for a vector of AOs as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out1' are the same	-
		as in 'atvec' 2) Check that the	
		number of elements in 'out2' 3)	
		Check that each output AO con-	
		tains the correct data.	
	Tests that the search method	Test that the search method	pass
03	works with a matrix of AOs as	works for a matrix of AOs as in-	P
	input.	put.	
	- mp attr	1) Check that the number of el-	pass
		ements in 'out1' are the same	Pass
		as in 'atmat' 2) Check that the	
		number of elements in 'out?' 3)	
		Check that each output AO con-	
		tains the correct data	
	Tests that the search method	Test that the search method	nass
04	works with a list of $\Delta \Omega_s$ as	works for a list of $\Delta \Omega_s$ as input	pass
	input	1) Check that the number of el-	nage
	input.	ements in 'out1' are the same	pass
		as the input 2) Check that the	
		number of elements in 'out?' 3)	
		Check that each output AO con	
		the correct data	
	Tosts that the search method	Test that the search method	Doca
05	works with a mix of different	works with an input of matrices	pass
	shaped AOs as input	and works with an input of matrices	
	shaped AOs as input.	1) Check that the number of al	naga
		1) Check that the number of el-	pass
		ements in out1 are the same	
		as the input 2) Uneck that the	
		number of elements in $[out2, 3]$	
		Check that 'out3' doesn't contain	
		an AO 4) Check that each output	
		AU contains the correct data.	
06	Iests that the search method	1 lest that the result of applying	pass
	properly applies history.	the search method can be pro-	
		cessed back.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/search			
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'search'. 2) Check that the re-	
		built object is the same object as	
		the input.	
07	Tests that the modifier call for	Tests that the modifier call for	pass
07	the search method doesn't work.	the search method doesn't work.	
			pass
08	Check that the search method	Call the method with a list of	pass
08	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 111: Unit tests for ao/search.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/select			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the select method works with a vector of AOs as input.	Test that the select method works for a vector of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atvec' 2) Check that each output AO contains the correct data.	pass
03	Tests that the select method works with a matrix of AOs as input.	Test that the select method works for a matrix of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atmat' 2) Check that each output AO contains the correct data.	pass
04	Tests that the select method works with a list of AOs as input.	 Test that the select method works for a list of AOs as input. 1) Check that the number of elements in 'out' is the same as in input. 2) Check that each output AO contains the correct data 	pass pass
05	Tests that the select method works with a mix of different shaped AOs as input.	Test that the select method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the select method properly applies history.	Test that the result of applying the select method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'select'. 2) Check that the re- built object is the same object as the input.	pass
07	Tests that the select method can modify the input AO.	Test that the select method can modify the input AO by call- ing with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/select			
		1) Check that 'out' and 'aeq' are now different. 2) Check that 'aeq' is not changed 3) Check that the modified input is the selected value of the copy 4) Check that out and amodi are the same	pass
08	Test the shape of the output.	 Test that the select method keeps the data shape of the input object. The input AO must be an AO with row data and an AO with column data. 1) Check that the shape of the 	pass
09	Check that the select method pass back the output objects to a list of output variables or to a single variable.	data doesn't change. Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
10	Test that the select method uses the plist to select the samples.	Test that the select method uses the plist to select the samples. 1) Check that the select method uses the samples in the plist 2) Check that the select method uses the samples in the plist and input 3) Check that the select method uses the samples in the plist and input 4) Check that the re-built object is the same as in 'out[13]'.	pass
11	Check that the [ao/select] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output. 1) Check that the output con-	pass
12	Test that the select method also select the 'dx' and 'dy' values.	 tains the same plotinfo plist Test that the select method also select the 'dx' and 'dy' values. 1) Check the output 2) Check that the re-built object is the same as in 'out'. 	pass pass
13	Test that the select method updates the t0.	Test that the select method up- dates the t0.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/select		
	1) Check the output 2) Check	pass
	that the re-built object is the	
	same as in 'out'.	

Table 112: Unit tests for ao/select.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/setDescription			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the setDescription method works with a vector of AOs as input.	Test that the setDescription method works for a vector of AOs as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atvec' 2) Check that each output AO contains the correct data.	pass
03	Tests that the setDescription method works with a matrix of AOs as input.	Test that the setDescription method works for a matrix of AOs as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atmat' 2) Check that each output AO contains the correct data.	pass
04	Tests that the setDescription method works with a list of AOs as input.	Test that the setDescription method works for a list of AOs as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
05	Tests that the setDescription method works with a mix of different shaped AOs as input.	Test that the setDescription method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the setDescription method properly applies history and that the option 'internal'	Test that the result of applying the setDescription method can be processed back to an m-file.	pass
	suppresses the history.	1) Check that the last entry in the history of 'out1' corresponds to 'setDescription'. 1) Check that the last entry in the history of 'out2' NOT corresponds to 'set- Description'. 2) Check that the re-built object is the same object as 'out'.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/setDescription			
07	Tests that the setDescription method can modify the input ΔO	Test that the setDescription method can modify the input AO by calling with no output	pass
	10.	 by caring with no output. Check that 'at1' and 'ain' are now different. 2) Check that 'ain' has the correct description field 	pass
08	Tests that the setDescription method can set the property with a plist.	Test that the setDescription method can modify the property 'description' with a value in a plist.	pass
		1) Check that 'ain' has the correct description field 2) Check that the re-built object is the same object as 'out'.	pass
09	Check that the setDescription method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
11	Check that the [ao/setDescription] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the same plotinfo plist	pass

Table 113: Unit tests for ao/setDescription.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/setFs			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the setFs method works with a vector of AOs as input.	Test that the setFs method works for a vector of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'avec' 2) Check that each output AO contains the correct data.	pass
03	Tests that the setFs method works with a matrix of AOs as input.	Test that the setFs method works for a matrix of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'amat' 2) Check that each output AO contains the correct data.	pass
04	Tests that the setFs method works with a list of AOs as	Test that the setFs method works for a list of AOs as input.	pass
	input.	1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
05	Tests that the setFs method works with a mix of different shaped AOs as input.	Test that the setFs method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the setFs method properly applies history and that the option 'internal'	Test that the result of applying the setFs method can be pro- cessed back to an m-file.	pass
	suppresses the history.	1) Check that the last entry in the history of 'out1' corresponds to 'setFs'. 1) Check that the last entry in the history of 'out2' NOT corresponds to 'setFs'. 2) Check that the re-built object is the same object as 'out'.	pass
07	Tests that the setFs method can modify the input AO.	Test that the setFs method can modify the input AO by calling with no output.	pass
		1) Check that 'at1' and 'ain' are now different. 2) Check that 'ain' has the correct fs field	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/setFs			
08	Tests that the setFs method can set the property with a plist.	Test that the setFs method can modify the property 'fs' with a value in a plist.	pass
		1) Check that 'ain' has the correct fs field 2) Check that the re- built object is the same object as 'out'.	pass
09	Check that the setFs method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
11	Check that the [ao/setFs] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the same plotinfo plist	pass

Table 114: Unit tests for ao/setFs.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/setName			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the setName method	Test that the setName method	pass
02	works with a vector of AOs as	works for a vector of AOs as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'atvec' 2) Check that each output	
		contains the correct data.	
03	Tests that the setName method	Test that the setName method	pass
00	works with a matrix of AOs as	works for a matrix of AOs as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'atmat' 2) Check that each out-	
		put contains the correct data.	
04	Tests that the setName method	Test that the setName method	pass
04	works with a list of AOs as	works for a list of AOs as input.	
	input.	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		contains the correct data.	
05	Tests that the setName method	Test that the setName method	pass
	works with a mix of different	works with an input of matrices	
	shaped AOs as input.	and vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		contains the correct data.	
06	Tests that the setName method	Test that the result of applying	pass
	properly applies history and	the setName method can be pro-	
	that the option 'internal'	cessed back to an m-file.	
	suppresses the history.	1) Check that the last entry in	pass
		the history of 'out1' corresponds	
		to 'setName'. 1) Check that the	
		last entry in the history of 'out2'	
		NOT corresponds to 'setName'.	
		2) Check that the re-built object	
		is the same object as 'out'.	
07	Tests that the setName method	Test that the setName method	pass
	can modify the input AO.	can modify the input AO by call-	
		ing with no output.	
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct name field	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/setName			
08	Tests that the setName method can set the property with a plict	Test that the setName method can modify the property 'name' with a value in a plict	pass
	pust.	1) Check that 'ain' has the cor	naga
		1) Check that all has the col-	pass
		rect name neid 2) Check that the	
		re-built object is the same object	
		as out.	
09	Check that the setName method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/setName]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 115: Unit tests for ao/setName.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/setPlotinfo			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the setPlotinfo	Test that the setPlotinfo method	pass
02	method works with a vector of	works for a vector of AOs as in-	
	AOs as input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'atvec' 2) Check that each output	
		AO contains the correct data.	
0.2	Tests that the setPlotinfo	Test that the setPlotinfo method	pass
03	method works with a matrix of	works for a matrix of AOs as in-	
	AOs as input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as	
		in 'atmat' 2) Check that each	
		output AO contains the correct	
		data.	
0.1	Tests that the setPlotinfo	Test that the setPlotinfo method	pass
04	method works with a list of AOs	works for a list of AOs as input.	-
	as input.	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
05	Tests that the setPlotinfo	Test that the setPlotinfo method	pass
05	method works with a mix of	works with an input of matrices	
	different shaped AOs as input.	and vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
06	Tests that the setPlotinfo	Test that the result of applying	pass
00	method properly applies history	the setPlotinfo method can be	
	and that the option 'internal'	processed back to an m-file.	
	suppresses the history.	1) Check that the last entry in	pass
		the history of 'out1' corresponds	
		to 'setPlotinfo'. 1) Check that	
		the last entry in the history of	
		'out2' NOT corresponds to 'set-	
		Plotinfo'. 2) Check that the re-	
		built object is the same object as	
		out'.	
07	Tests that the setPlotinfo	Test that the setPlotinfo method	pass
07	method can modify the input	can modify the input AO by call-	.
	AO.	ing with no output.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/setPlotinfo			
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct plotinfo field	
08	Tests that the setPlotinfo	Test that the setPlotinfo method	pass
00	method can set the property	can modify the property 'plot-	
	with a plist.	info' with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect plotinfo field 2) Check that	
		the re-built object is the same	
		object as 'out'.	
00	Check that the setPlotinfo	Call the method with a list of	pass
09	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 116: Unit tests for ao/setPlotinfo.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/setT0			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		baye the correct parameters	
	Tests that the setT0 method	Test that the setT0 method	nass
02	works with a vector of AOs as	works for a vector of AOs as in-	pass
	input.	put.	
	1	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'avec' 2) Check that each output	
		AO contains the correct data.	
03	Tests that the setT0 method	Test that the setT0 method	pass
	works with a matrix of AOs as	works for a matrix of AOs as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		$\Delta \Omega$ contains the correct data	
	Tests that the setT0 method	Test that the setT0 method	nage
04	works with a list of AOs as	works for a list of AOs as input	pass
	input.	1) Check that the number of el-	pass
	L	ements in 'out' is the same as in	1
		input. 2) Check that each output	
		AO contains the correct data.	
05	Tests that the setT0 method	Test that the setT0 method	pass
00	works with a mix of different	works with an input of matrices	
	shaped AOs as input.	and vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		AQ contains the correct date	
	Tests that the setT0 method	Tost that the result of applying	nass
06	properly applies history and	the setT0 method can be pro-	pass
	that the option 'internal'	cessed back to an m-file.	
	suppresses the history.	1) Check that the last entry in	pass
		the history of 'out1' corresponds	1
		to 'setT0'. 1) Check that the	
		last entry in the history of 'out2'	
		NOT corresponds to 'setT0'. 2)	
		Check that the re-built object is	
		the same object as 'out'.	
07	Tests that the setT0 method	Test that the setT0 method can	pass
	can modify the input AO.	modify the input AO by calling	
		with no output.	
		1) Uneck that 'atl' and 'ain' are	pass
		has the correct to field	
1		I has the correct to held	1



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/setT0			
08	Tests that the setT0 method can set the property with a plist.	Test that the setT0 method can modify the property 't0' with a value in a plist.	pass
		1) Check that 'ain' has the correct t0 field 2) Check that the re- built object is the same object as 'out'.	pass
09	Check that the setT0 method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
11	Check that the [ao/setT0] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the same plotinfo plist	pass

Table 117: Unit tests for ao/setT0.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/setX			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the setX method works with a vector of AOs as input.	Test that the setX method works for a vector of AOs as input. 1) Check that the number of el-	pass pass
		ements in 'out' is the same as in 'avec' 2) Check that each output AO contains the correct data.	
03	Tests that the setX method works with a matrix of AOs as	Test that the setX method works for a matrix of AOs as input.	pass
	mput.	1) Check that the number of el- ements in 'out' is the same as in 'amat' 2) Check that each output AO contains the correct data.	pass
04	Tests that the setX method works with a list of AOs as	Test that the setX method works for a list of AOs as input.	pass
	input.	1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
05	Tests that the setX method works with a mix of different shaped AOs as input.	Test that the setX method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the setX method properly applies history and that the option 'internal'	Test that the result of apply- ing the setX method can be pro- cessed back to an m-file.	pass
	suppresses the history.	1) Check that the last entry in the history of 'out1' corresponds to 'setX'. 1) Check that the last entry in the history of 'out2' NOT corresponds to 'setX'. 2) Check that the re-built object is the same object as 'out'.	pass
07	Tests that the setX method can modify the input AO.	Test that the setX method can modify the input AO by calling with no output.	pass
		1) Check that 'at1' and 'ain' are now different. 2) Check that 'ain' has the correct 'x' field	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/setX			
08	Tests that the setX method can	Test that the setX method can	pass
08	set the property with a plist.	modify the property 'x' with a	
		value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect x field 2) Check that the re-	
		built object is the same object as	
		'out'.	
00	Tests that the setX method	Test that the setX method keeps	pass
09	keeps the shape of the data.	the shape of the data. Indepen-	
		dent of the input	
		1) Check that the shape of the	pass
		data doesn't change.	
10	Check that the setX method	Call the method with a list of	pass
10	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/setX]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 118: Unit tests for ao/setX.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/setXY			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set	pass
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the setXY method	Test that the setXY method	pass
02	works with a vector of AOs as	works for a vector of AOs as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'avec' 2) Check that each output	
		AO contains the correct data.	
0.2	Tests that the setXY method	Test that the setXY method	pass
03	works with a matrix of AOs as	works for a matrix of AOs as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'amat' 2) Check that each output	
		AO contains the correct data.	
0.4	Tests that the setXY method	Test that the setXY method	pass
04	works with a list of AOs as	works for a list of AOs as input.	
	input.	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
05	Tests that the setXY method	Test that the setXY method	pass
05	works with a mix of different	works with an input of matrices	
	shaped AOs as input.	and vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
06	Tests that the setXY method	Test that the result of applying	pass
00	properly applies history and	the setXY method can be pro-	
	that the option 'internal'	cessed back to an m-file.	
	suppresses the history.	1) Check that the last entry in	pass
		the history of 'out1' corresponds	
		to 'setXY'. 1) Check that the	
		last entry in the history of 'out2'	
		NOT corresponds to 'setXY'. 2)	
		Check that the re-built object is	
		the same object as 'out'.	
07	Tests that the setXY method	Test that the setXY method can	pass
	can modify the input AO.	modify the input AO by calling	
		with no output.	
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct 'x' and 'y' field	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/setXY			
08	Tests that the setXY method	Test that the setXY method can	pass
08	can set the property with a	modify the property 'x' and 'y'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect x and y fields 2) Check that	
		the re-built object is the same	
		object as 'out'.	
00	Tests that the setXY method	Test that the setXY method	pass
09	keeps the shape of the data.	keeps the shape of the data. In-	
		dependent of the input	
		1) Check that the shape of the	pass
		data doesn't change.	
10	Check that the setXY method	Call the method with a list of	pass
10	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/setXY]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 119: Unit tests for a o/setXY.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/setXunits			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the setXunits method	Test that the setXunits method	pass
02	works with a vector of AOs as	works for a vector of AOs as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'avec' 2) Check that each output	
		AO contains the correct data.	
0.2	Tests that the setXunits method	Test that the setXunits method	pass
03	works with a matrix of AOs as	works for a matrix of AOs as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'amat' 2) Check that each output	
		AO contains the correct data.	
	Tests that the setXunits method	Test that the setXunits method	pass
04	works with a list of AOs as	works for a list of AOs as input.	1
	input.	1) Check that the number of el-	pass
	I. and	ements in 'out' is the same as in	1
		input. 2) Check that each output	
		AO contains the correct data.	
	Tests that the setXunits method	Test that the setXunits method	pass
05	works with a mix of different	works with an input of matrices	1
	shaped AOs as input.	and vectors and single AOs.	
	I I I I I I I I I I I I I I I I I I I	1) Check that the number of el-	pass
		ements in 'out' is the same as in	1
		input, 2) Check that each output	
		AQ contains the correct data.	
	Tests that the setXunits method	Test that the result of apply-	pass
06	properly applies history and	ing the setXunits method can be	pass
	that the option 'internal'	processed back to an m-file	
	suppresses the history.	1) Check that the last entry in	pass
	suppresses the instory.	the history of 'out1' corresponds	Pass
		to 'setXunits' 1) Check that the	
		last entry in the history of 'out?'	
		NOT corresponds to 'setXunits'	
		2) Check that the re-built object	
		is the same object as 'out'	
	Tests that the setXunits method	Test that the setXunits method	nass
07	can modify the input AO	can modify the input AO by call	Pass
	an mouny the input AO.	ing with no output	
		1) Check that 'at1' and 'ain' are	nase
		now different 2) Check that 'sin'	pass
		has the correct munite fold	
		has the correct xunits field	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/setXunits			
08	Tests that the setXunits method can set the property with a	Test that the setXunits method can modify the property 'xunits'	pass
	plist.	with a value in a plist.	
	-	1) Check that 'ain' has the cor-	pass
		rect xunits field 2) Check that	
		the re-built object is the same	
		object as 'out'.	
00	Check that the setXunits	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/setXunits]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 120: Unit tests for ao/setXunits.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/setY			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
	Tests that the setV method	nave the correct parameters.	
02	rests that the set Y method	for a vector of AOa as input	pass
	input	1) Check that the number of el	nase
	mput.	ements in 'out' is the same as in	pass
		'atvec' 2) Check that each output	
		AO contains the correct data	
	Tests that the setY method	Test that the setY method works	pass
03	works with a matrix of AOs as	for a matrix of AOs as input.	pass
	input.	1) Check that the number of el-	pass
	r	ements in 'out' is the same as	1
		in 'atmat' 2) Check that each	
		output AO contains the correct	
		data.	
0.4	Tests that the setY method	Test that the setY method works	pass
04	works with a list of AOs as	for a list of AOs as input.	
	input.	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
05	Tests that the setY method	Test that the setY method works	pass
	works with a mix of different	with an input of matrices and	
	shaped AOs as input.	vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AU contains the correct data.	
06	rests that the set Y method	ing the setV method can be pro-	pass
	that the option 'internal'	cossed back to an m file	
	suppresses the history	1) Check that the last entry in	nase
	suppresses the instory.	the history of 'out1' corresponds	pass
		to 'setY' 1) Check that the last	
		entry in the history of 'out?'	
		NOT corresponds to 'setY'. 2)	
		Check that the re-built object is	
		the same object as 'out'.	
07	Tests that the setY method can	Test that the setY method can	pass
07	modify the input AO.	modify the input AO by calling	1
		with no output.	
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct 'y' field	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/setY			
08	Tests that the setY method can	Test that the setY method can	pass
00	set the property with a plist.	modify the property 'y' with a	
		value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect y field 2) Check that the re-	
		built object is the same object as	
		'out'.	
00	Tests that the setY method	Test that the setY method keeps	pass
09	keeps the shape of the data.	the shape of the data. Indepen-	
		dent of the input	
		1) Check that the shape of the	pass
		data doesn't change.	
10	Check that the setY method	Call the method with a list of	pass
10	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/setY]	Call the method with a list of	pass
11	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 121: Unit tests for ao/setY.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/setYunits			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the setYunits method works with a vector of AOs as input.	Test that the setYunits method works for a vector of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atvec' 2) Check that each output AO contains the correct data.	pass
03	Tests that the setYunits method works with a matrix of AOs as input.	Test that the setYunits method works for a matrix of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'atmat' 2) Check that each output AO contains the correct data.	pass
04	Tests that the setYunits method works with a list of AOs as	Test that the setYunits method works for a list of AOs as input.	pass
	input.	1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
05	Tests that the setYunits method works with a mix of different shaped AOs as input.	Test that the setYunits method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the setYunits method properly applies history and that the option 'internal'	Test that the result of applying the setYunits method can be pro- cessed back to an m-file.	pass
	suppresses the history.	 Check that the last entry in the history of 'out1' corresponds to 'setYunits'. 1) Check that the last entry in the history of 'out2' NOT corresponds to 'setYunits'. Check that the re-built object is the same object as 'out'. 	pass
07	Tests that the setYunits method can modify the input AO.	Test that the setYunits method can modify the input AO by call- ing with no output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/setYunits			
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct yunits field	
08	Tests that the setYunits method	Test that the setYunits method	pass
08	can set the property with a	can modify the property 'yunits'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect yunits field 2) Check that	
		the re-built object is the same	
		object as 'out'.	
00	Check that the setYunits	Call the method with a list of	pass
09	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/setYunits]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 122: Unit tests for ao/setYunits.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/sign			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		1) Check that got Info call ro	neg
		turned an minfo object in all	pass
		cases 2) Check that all plists	
		have the correct parameters	
	Tosts that the [ac/sign] method	Tost that the [ao/sign] method	nase
02	works with a voctor of objects	works for a voctor of objects as	pass
	as input	input	
	as input.	1) Check that the number of el-	nass
		ements in 'out' is the same as in	pass
		'vec' 2) Check that each output	
		object contains the correct data	
	Tests that the [ao/sign] method	Test that the [ao/sign] method	nass
03	works with a matrix of objects	works for a matrix of objects as	pass
	as input	input	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	P
		'mat' 2) Check that each output	
		object contains the correct data.	
	Tests that the [ao/sign] method	Test that the [ao/sign] method	pass
04	works with a list of objects as	works for a list of objects as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/sign] method	Tests that the [ao/sign] method	pass
00	works with a mix of different	works with a mix of different ar-	
	arrays of objects as input.	rays of objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
06	Tests that the [ao/sign] method	Test that the result of applying	pass
	properly applies history.	the [ao/sign] method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[ao/sign]'. 2) Check that the	
		re-built object is the same object	
		as the input.	
07	Tests that the [ao/sign] method	I lest that the [ao/sign] method	pass
	can modify the input AO.	can modify the input object by	
		calling with no output and that	
		input of the function met ti	
		input of the function notation	
		(with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/sign			
		1) Check that 'out' and 'aeq' are now different. 2) Check that 'aeq' is not changed 3) Check that the modified input is the [ao/sign] value of the copy 4) Check that out and amodi are the same	pass
09	Test the shape of the data in AOs.	Test that the [ao/sign] method keeps the data shape of the in- put object. The input AO data must be an array with row data and/or column data.	pass
		1) Check that the shape of the data doesn't change.	pass
10	Check that the [ao/sign] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
11	Check that the [ao/sign] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the same plotinfo plist	pass

Table 123: Unit tests for a o/sign.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/simplifyYunits			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the simplifyYunits	Test that the simplifyYunits	pass
02	method works with a vector of	method works for a vector of AOs	
	AOs as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'atvec' 2) Check that each output	
		AO contains the correct data.	
03	Tests that the simplifyYunits	Test that the simplifyYunits	pass
00	method works with a matrix of	method works for a matrix of	
	AOs as input.	AOs as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as	
		in 'atmat' 2) Check that each	
		output AO contains the correct	
		data.	
04	Tests that the simplifyYunits	Test that the simplifyYunits	pass
04	method works with a list of AOs	method works for a list of AOs	
	as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
05	Tests that the simplifyYunits	Test that the simplifyYunits	pass
00	method works with a mix of	method works with an input of	
	different shaped AOs as input.	matrices and vectors and single	
		AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
06	Tests that the simplifyYunits	Test that the result of applying	pass
	method properly applies history.	the simplifyYunits method can	
		be processed back.	
		1) Check that the last entry	pass
		in the history of 'out' corre-	
		sponds to 'simplifyYunits'. 2)	
		Check that the re-built object is	
		the same object as the input.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/simplifyYunits			
07	Tests that the simplifyYunits	Test that the simplifyYunits	pass
	method can modify the input	method can modify the input AO	
	AO.	by calling with no output and	
		that the method doesn't change	
		the input of the function nota-	
		tion (with a equal sign).	
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	T
		is not changed 3) Check that the	
		modified input simplifies the v-	
		units of the copy 4) Check that	
		out and amodi are the same	
	Check that the simplify Vunits	Call the method with a list of	negg
08	method pass back the output	output variables and with a sin	pass
	abjects to a list of output	de output variable. Additionaly	
	variables or to a single variable	sheak that the rebuild method	
	variables of to a single variable.	works on the output	
		1) Check that the output con	nass
		toing the right number of ob	pass
		ioets 2) Check that the 'rebuild'	
		method produces the same of	
		inethod produces the same ob-	
	Test that the simplify Vunite	Test that the simplify Vunite	naga
09	works for different complex	works for different complex y	pass
	works for different complex	works for different complex y-	
	y-units.	1) Check that the connect data 2)	nagg
		Check that the correct data 2)	pass
		the same as in 'out[1, 3]'	
	Test that the simplify Vunite	Test that the simplify Vunite	naga
10	works with a sweeption list	works with a avaantion list	pass
	works with a exception list.	1) Check that the correct date 2)	magg
		Check that the correct data 2)	pass
		the same as in 'out[1, 2]'	
	Cheels that the	the same as in $\operatorname{Out}[12]$.	magg
11	oneck that the	output variables and with a sin	pass
	[ao/simpiny funits] method pass	output variables and with a sin-	
	f autout and include a simple	gle output variable. Additionally	
	of output variables or to a single	check that the rebuild method	
	vanable.	1) Check that the extruct	Decc
		1) Oneck that the output con-	pass
		tains the same plotinio plist	
12	Lest that the simplify Yunits	lest that the simplify Yunits	pass
	doesn't apply the prefixes to the	doesn't apply the prefixes to the	
	data.	data.	
		1) Uneck that the correct data 2)	pass
		Check that the re-built object is	
		the same as in $[out[12]]$.	

Table 124: Unit tests for ao/simplifyYunits.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/sin			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the [ao/sin] method	Test that the [ao/sin] method	pass
02	works with a vector of objects	works for a vector of objects as	
	as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'vec' 2) Check that each output	
		object contains the correct data.	
03	Tests that the [ao/sin] method	Test that the [ao/sin] method	pass
00	works with a matrix of objects	works for a matrix of objects as	
	as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
04	Tests that the [ao/sin] method	Test that the [ao/sin] method	pass
04	works with a list of objects as	works for a list of objects as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/sin] method	Tests that the [ao/sin] method	pass
	works with a mix of different	works with a mix of different ar-	
	arrays of objects as input.	rays of objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
06	Tests that the [ao/sin] method	Test that the result of applying	pass
	properly applies history.	the [ao/sin] method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[ao/sin]'. 2) Check that the	
		re-built object is the same object	
		as the input.	
07	Tests that the [ao/sin] method	Test that the [ao/sin] method	pass
	can modify the input AO.	can modify the input object by	
		calling with no output and that	
		the method doesn't change the	
		input of the function notation	
		(with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/sin			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	-
		is not changed 3) Check that	
		the modified input is the [ao/sin]	
		value of the copy 4) Check that	
		out and amodi are the same	
	Test that the [ao/sin] method	Test that the [ao/sin] method	pass
08	uses the plist to get the axis.	uses the plist to get the axis.	-
		1) Check that the [ao/sin]	pass
		method applies to the x-axis 2)	-
		Check that the [ao/sin] method	
		applies to the y-axis 3) Check	
		that the [ao/sin] method applies	
		to both axes 4) Check that the	
		re-built object is the same as in	
		'out[13]'.	
00	Test the shape of the data in	Test that the [ao/sin] method	pass
09	AOs.	keeps the data shape of the in-	
		put object. The input AO data	
		must be an array with row data	
		and/or column data.	
		1) Check that the shape of the	pass
		data doesn't change.	
10	Check that the [ao/sin] method	Call the method with a list of	pass
10	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/sin] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 125: Unit tests for a o/sin.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/smoother			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the smoother method works with a vector of AOs as input.	Test that the smoother method works for a vector of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
03	Tests that the smoother method works with a matrix of AOs as input.	Test that the smoother method works for a matrix of AOs as in- put.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
04	Tests that the smoother method works with a list of AOs as	Test that the smoother method works for a list of AOs as input.	pass
	mpuv.	ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	Pass
05	Tests that the smoother method works with a mix of different shaped AOs as input.	Test that the smoother method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the smoother method properly applies history.	Test that the result of applying the smoother method can be pro- cessed back to an m-file.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'smoother'. 2) Check that the re-built object is the same object as 'out'.	pass
07	Tests that the smoother method can modify the input AO.	Test that the smoother method can modify the input AO by call- ing with no output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/smoother			
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		is smoother(at5).	
08	Tests that the smoother method	Test that the smoother method	pass
00	keeps the data shape of the	keeps the data shape of the in-	
	input object.	put object. The input AO must	
		be an AO with row data and an	
		AO with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
00	Check that the smoother	Call the method with a list of	pass
09	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/smoother]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 126: Unit tests for ao/smoother.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/sort			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually. 1) Check that getInfo call re- turned an minfo object in all cases 2) Check that all pliets	pass pass
		have the correct parameters.	
02	Tests that the sort method works with a vector of AOs as input.	Test that the sort method works for a vector of AOs as input. 1) Check that the number of el- ements in 'out' is the same as in 'atvec' 2) Check that each output AO contains the correct data.	pass
03	Tests that the sort method works with a matrix of AOs as input.	Test that the sort method works for a matrix of AOs as input. 1) Check that the number of el- ements in 'out' is the same as in 'atmat' 2) Check that each output AO contains the correct data.	pass pass
04	Tests that the sort method works with a list of AOs as input.	Test that the sort method works for a list of AOs as input. 1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
05	Tests that the sort method works with a mix of different shaped AOs as input.	 Test that the sort method works with an input of matrices and vectors and single AOs. 1) Check that the number of elements in 'out' is the same as in input. 2) Check that each output AO contains the correct data. 	pass
06	Tests that the sort method properly applies history.	Test that the result of applying the sort method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'sort'. 2) Check that the re- built object is the same object as the input.	pass
07	Tests that the sort method can modify the input AO.	Test that the sort method can modify the input AO by call- ing with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/sort			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the sort value of	
		the copy 4) Check that out and	
		amodi are the same	
0.0	Trat the share of the cost and	Test that the sort method keeps	pass
08	lest the snape of the output.	the data shape of the input ob-	
		ject. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shape of the	pass
		data doesn't change.	1
	Check that the sort method	Call the method with a list of	pass
09	pass back the output objects to	output variables and with a sin-	1
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	Pass
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'	
	Test that the sort method uses	Test that the sort method uses	nass
10	the plist to sort on the y- or y-	the plist to get the axis	pass
	avis	1) Check that the sort method	nass
		applies to the y-axis in ascend	pass
		direction 2) Check that the sort	
		method applies to the v-axis in	
		descend direction 3) Check that	
		the sort method applies to the x_{-}	
		axis in ascend direction 4) Check	
		that the sort method applies to	
		that the soft method applies to the x_{-} axis in descend direction 5)	
		Check that the re built object is	
		the same as in 'out[1 4]'	
	Check that the [ao/sort] method	Call the method with a list of	nase
11	pass back the output objects to	output variables and with a sin	pass
	a list of output variables or to a	ale output variable. Additional	
	a list of output variables of to a	sheels that the population and	
	single variable.	works on the output	
		1) Check that the extruct	Decc
		1) Uneck that the output con-	pass
		tains the same piotinio plist	

Table 127: Unit tests for ao/sort.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/spectrogram			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the spectrogram method works with a vector of AOs as input.	Test that the spectrogram method works for a vector of AOs as input.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
03	Tests that the spectrogram method works with a matrix of AOs as input.	Test that the spectrogram method works for a matrix of AOs as input.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
04	Tests that the spectrogram method works with a list of AOs as input.	Test that the spectrogram method works for a list of AOs as input.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
05	Tests that the spectrogram method works with a mix of different shaped AOs as input.	Test that the spectrogram method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the spectrogram method properly applies history.	Test that the result of applying the spectrogram method can be processed back to an m-file.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'spectrogram'. 2) Check that the re-built object is the same object as 'out'.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/spectrogram			
11	Check that the [ao/spectrogram]	Call the method with a list of	pass
11	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 128: Unit tests for ao/spectrogram.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

split/chunks			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the split method works with a vector of AOs as input.	Test that the split method works for a vector of AOs as input. 1) Check that the number of out-	pass pass
	•	puts 2) Check that each output AO contains the correct data.	
03	Tests that the split method works with a matrix of AOs as input.	Tests that the split method works with a matrix of AOs as input.	pass
		1) Check that the number of outputs 2) Check that each output AO contains the correct data.	pass
04	Tests that the split method works with a list of AOs as input.	Tests that the split method works with a matrix of AOs as input.	pass
		1) Check that the number of out- puts 2) Check that each output AO contains the correct data.	pass
05	Tests that the split method works with a mix of different shaped AOs as input.	Tests that the split method works with a matrix of AOs as input.	pass
		1) Check that the number of out- puts 2) Check that each output AO contains the correct data.	pass
06	Tests that the split method properly applies history.	Test that the result of apply- ing the split method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'split'. 2) Check that the re- built object is the same object as the input.	pass
07	The split method can not modify the input AO.	The split method can not modify the input AO. 1) Nothind to do.	pass
08	Test the shape of the output.	Test that the split method keeps the data shape of the input ob- ject. The input AO must be an AO with row data and an AO with column data.	pass
		data doesn't change.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

split/chunks			
09	Check that the split method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the rebuild	
		method produces the same ob-	
	Chash that the galit method	Check that the galit method as	D 0 0 0
10	Check that the split method	Check that the split method ac-	pass
	'accepts both key-words in and	'cepts both key-words in and	
	Chunks .	1) Check that the number of out	naga
		nuts 2) Check that the number of out-	pass
		method produces the same ob-	
		iect as 'out'	
	Check that the [split/chunks]	Call the method with a list of	pass
11	method pass back the output	output variables and with a sin-	Pass
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	-
10	Tests that the split method	Tests that the split method	pass
12	works also split 'dx' and 'dy'.	works also split 'dx' and 'dy'.	
		1) Check that the number of out-	pass
		puts 2) Check that each output	
		AO contains the correct data.	
13	Check that the split method	Check that the split method ig-	pass
10	ignores the samples which	nores the samples which doesn't	
	doesn't fit.	fit. Use for this test the key-word	
		'match'	
		1) Check that the number of out-	pass
		puts 2) Check that the 'rebuild'	
		method produces the same ob-	
	Choole that the calit method	Choole that the colit most of	near
15	works with notocusl sampled	works with notocycl compled	pass
	data Check that the split	data Chock that the calit	
	method works with 'timoshift'	method works with 'timoshift'	
	option	option	
		1) Check the outputs 2) Check	nase
		that the 'rebuild' method pro-	Pass
		duces the same object as 'out'	
		that the 'rebuild' method pro- duces the same object as 'out'.	

Table 129: Unit tests for split/chunks.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

split/interval			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		turned an minfo object in all cases. 2) Check that all plists	pass
	Tosta that the split method	Togt that the split method works	page
02	works with a vector of AOs as	for a vector of AOs as input	pass
	input.	1) Check that the number of el-	pass
	input.	ements in 'out' is the same as in	pass
		'avec' times numbers of intervals	
		2) Check that each output AO	
		contains the correct data.	
0.2	Tests that the split method	Tests that the split method	pass
03	works with a matrix of AOs as	works with a matrix of AOs as	
	input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'amat' times numbers of intervals	
		2) Check that each output AO	
		contains the correct data.	
04	Tests that the split method	Tests that the split method	pass
	works with a list of AOs as	works with a list of AOs as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		2) Check that each autout AO	
		2) Check that each output AO	
	Tests that the split method	Tests that the split method	nass
05	works with a mix of different	works with a mix of different	pass
	shaped AOs as input	shaped AOs as input	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	1
		'amat' times numbers of intervals	
		2) Check that each output AO	
		contains the correct data.	
06	Tests that the split method	Test that the result of apply-	pass
00	properly applies history.	ing the split method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'split'. 2) Check that the re-	
		built object is the same object as	
		the input.	
07	The split method can not	The split method can not modify	pass
	modify the input AO.	the input AO.	
		1) Nothing to do.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

split/interval			
08	Test the shape of the output.	Test that the split method keeps the data shape of the input ob- ject. The input AO must be an AO with row data and an AO with column data. 1) Check that the shape of the data doesn't change.	pass
09	Check that the split method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
10	Check that the split method can handle AO which have a collaped x-axis.	Check that the split method can handle AO which have a collaped x-axis.	pass
		 Check the number of outputs Check the output 3) Check that the 'rebuild' method pro- duces the same object as 'out'. 	pass
11	Check that the [split/interval] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the same plotinfo plist	pass
12	Check that the split method uses the key-words: start_time and duration	Check that the split method uses the key-words: start_time and duration	pass
		 Check the number of outputs Check the output 3) Check that the 'rebuild' method pro- duces the same object as 'out'. 	pass
13	Check that the split method uses the key-words: timespan	Check that the split method uses the key-words: timespan	pass
		 Check the number of outputs Check the output 3) Check that the 'rebuild' method pro- duces the same object as 'out'. 	pass
14	Check that the split method uses the key-words: start_time and end_time	Check that the split method uses the key-words: start_time and end_time	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

split/interval			
		 Check the number of outputs Check the output 3) Check that the 'rebuild' method pro- duces the same object as 'out'. 	pass
15	Check that the split method works with notequal sampled data. Check that the split method works with 'timeshift' option. Check that the interval can be passed via two time objects.	Check that the split method works with notequal sampled data. Check that the split method works with 'timeshift' option. Check that the interval can be passed via two time objects. 1) Check the outputs 2) Check	pass
		that the 'rebuild' method pro- duces the same object as 'out'.	1
16	Check that the split method works with notequal sampled data. Check that the split method works with 'timeshift' option. Check that the interval can be passed via two time strings.	Check that the split method works with notequal sampled data. Check that the split method works with 'timeshift' option. Check that the interval can be passed via two time strings.	pass
		that the 'rebuild' method pro- duces the same object as 'out'.	разэ
17	Check that the split method works with notequal sampled data. Check that the split method works with 'timeshift' option. Check that the interval can be passed via one timespan object.	Check that the split method works with notequal sampled data. Check that the split method works with 'timeshift' option. Check that the interval can be passed via one timespan object.	pass
		1) Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'.	pass
18	Check that the split method works with notequal sampled data. Check that the split method works with 'timeshift' option. Check that the interval can be passed via one timespan object.	Check that the split method works with notequal sampled data. Check that the split method works with 'timeshift' option. Check that the interval can be passed via one timespan object.	pass
		that the 'rebuild' method pro- duces the same object as 'out'.	pass

Table 130: Unit tests for split/interval.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



split/samples			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists	pass
		nave the correct parameters.	<u> </u>
02	Tests that the split method works with a vector of AOs as	for a vector of AOs as input.	pass
	input.	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		atvect times numbers of inter-	
		Vals 2) Uneck that each output	
	Tests that the split method	AO contains the correct data.	naga
03	works with a matrix of $\Delta \Omega_{\rm S}$ as	works with a matrix of AOs as	pass
	input	input	
	input.	1) Check that the number of el-	pass
		ements in 'out' is the same as in	Pass
		'atmat' times numbers of inter-	
		vals 2) Check that each output	
		AO contains the correct data.	
0.4	Tests that the split method	Tests that the split method	pass
04	works with a list of AOs as	works with a list of AOs as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'atvec' times numbers of inter-	
		vals 2) Check that each output	
		AO contains the correct data.	
05	lests that the split method	lests that the split method	pass
	shaped AOs as input	works with a mix of different	
	snaped AOs as input.	1) Check that the number of el-	nase
		ements in 'out' is the same as in	pass
		'atvec' times numbers of inter-	
		vals 2) Check that each output	
		AO contains the correct data.	
00	Tests that the split method	Test that the result of apply-	pass
00	properly applies history.	ing the split method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'split'. 2) Check that the re-	
		built object is the same object as	
		the input.	<u> </u>
07	The split method can not	The split method can not modify	pass
	modify the input AO.	the input AO.	
		1) INOUMING TO GO.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

split/samples			
08	Tost the shape of the output	Test that the split method keeps	pass
00	Test the shape of the output.	the data shape of the input ob-	
		ject. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shape of the	pass
		data doesn't change.	
00	Check that the split method	Call the method with a list of	pass
09	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
10	Test that the split method also	Test that the split method also	pass
10	splits the 'dx' and 'dy' values.	splits the 'dx' and 'dy' values.	
		1) Check the output 2) Check	pass
		that the re-built object is the	
		same as in 'out'.	
11	Check that the [split/samples]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	
15	Check that the split method	Check that the split method	pass
10	works with notequal sampled	works with notequal sampled	
	data. Check that the split	data. Check that the split	
	method works with 'timeshift'	method works with 'timeshift'	
	option.	option.	
		1) Check the outputs 2) Check	pass
		that the 'rebuild' method pro-	
		duces the same object as 'out'.	

Table 131: Unit tests for split/samples.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



times/frequ			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists	pass
		have the correct parameters.	
02	Tests that the split method works with a vector of AOs as	Test that the split method works for a vector of AOs as input.	pass
	input.	1) Check that the number of el- ements in 'out' is the same as in	pass
		'atvec' times numbers of inter- vals 2) Check that each output	
	Tests that the split method	AO contains the correct data.	nass
03	works with a matrix of AOs as	works with a matrix of AOs as	pass
	input.	input. 1) Check that the number of el-	pass
		ements in 'out' is the same as in 'atway' times numbers of inter	
		vals 2) Check that each output	
		AO contains the correct data.	
04	Tests that the split method works with a list of AOs as	Tests that the split method works with a list of AOs as in-	pass
	input.	put.	
	1	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'atvec' times numbers of inter-	
		AO contains the correct data.	
05	Tests that the split method	Tests that the split method	pass
	shaped AOs as input.	shaped AOs as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in 'atway' times numbers of inter	
		vals 2) Check that each output	
		AO contains the correct data.	
06	Tests that the split method	Test that the result of apply-	pass
00	properly applies history.	ing the split method can be pro-	
		cessed back.	
		1) Check that the last entry in the history of 'out' corresponde	pass
		to 'split'. 2) Check that the re-	
		built object is the same object as	
		the input.	
07	The split method can not	The split method can not modify	pass
07	modify the input AO.	the input AO.	
		1) Nothind to do.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



08 Test the shape of the output. Test that the split method keeps the data shape of the input object and the input object. The input AO must be an AO with column data. Itest that the split method pass back the output variables and with a single variable. Itest that the split method with a list of pass back the output variables and with a single output variables and with a single output variables and with a single output variable. Itest that the rebuild method with a list of output variables and with a single output variable. 10 Check that the split method interprets a negative end interval as a indicates count from end and a zero for the and interval as the end of the vector. Ites that the split method works with notequal sampled data. Itest that the split method works on the output. pass back the output variable. 11 Check that the split method works or to a single variable. Check that the output contains the rebuild method produces the same object as 'out'. Itest that the split method works on the output. pass back the output variable. Additionaly check that the rebuild method works on the output. 11 Check that the split method also split 'dx' and 'dy'. Itest that the split method also split 'dx' and 'dy'. Itest the split method also split 'dx' and 'dy'. 13 Check that the split method interprets a negative end interval as the end of the vector. Itest that the 'rebuild' method produces the same object as 'out'. 14 Check that the split method interprets a negative end interval as the end of the vector. Itest that the 're	times/frequ			
10 Check that the split method pass back the output objects to a list of output variables or to a single variable. Call the method with a list of output variables and with a single output variables and with a single output variable. Additionally check that the rebuild method works on the output. 10 Check that the split method interprets a negative end interval as the end of the vector. Check that the split method interprets a negative end interval as the end of the vector. Check that the split method pass back the output objects to a list of output variables and with a single output variable. Additionally check that the rebuild method works on the output 2) Check that the split method also split 'dx' and 'dy'. 12 Check that the split method also split 'dx' and 'dy'. Check that the split method also split 'dx' and 'dy' if this values have the same length of the y-values. 13 Check that the split method interprets a negative end interval as the end of the vector. Check that the split method produces the same object as 'out'. 14 Check that the split me	08	Test the shape of the output.	Test that the split method keeps the data shape of the input ob- ject. The input AO must be an AO with row data and an AO with column data.	pass
09 Check that the split method pass back the output objects to a list of output variables or to a single variable. Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output. pa 10 Check that the split method interprets a negative end interval as a indicates count from end and a zero for the and interval as the end of the vector. Check this behavior for time- series data. pa 11 Check that the split method works on the output. Check that the filmes/frequ method pass back the output objects to a list of output variables or to a single variable. Check that the rebuild method works with notequal sampled data. Check that the split method works with notequal sampled data. 12 Check that the split method also split 'dx' and 'dy'. Check that the split method interval as the end of the vector. Check that the split method works with notequal sampled data. Check that the split method also split 'dx' and 'dy'. Check that the split method interval as a indicates count from end and a zero for the and interval as the end of the vector. Check that the split method also split 'dx' and 'dy'. Check that the split method also split 'dx' and 'dy'. Check that the split method interval as a indicates count from end and a zero for the and interval as the end of the vector. Check that the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'. 14 Check that the split method interval as a indicates count from end and a zero for the and interval as the end of the vector.<			1) Check that the shape of the data doesn't change.	pass
10Check that the split method interval as a indicates count from end and a zero for the and interval as the end of the vector.11) Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'.11Check that the [times/frequ] method pass back the output objects to a list of output variables or to a single variable.Call the method with a list of output variables and with a sin- gle output variable. Additionally check that the rebuild method works with notequal sampled data.12Check that the split method works with notequal sampled data.Check that the split method works with notequal sampled data.13Check that the split method interval as a indicates count from end and a zero for the and interval as a indicates count from end and a zero for the and interval as the end of the vector.Check that the split method works with notequal sampled data.14Check that the split method interval as a indicates count from end and a zero for the and interval as the end of the vector.Check that the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'.14	09	Check that the split method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
10 Check that the split method interprets a negative end interval as a indicates count from end and a zero for the and interval as the end of the vector. Check the output 2) Check that the 'rebuild' method produces the same object as 'out'. pa 11 Check that the [times/frequ] method pass back the output objects to a list of output variables or to a single variable. Call the method with a list of output variables and with a sin-object so a list of output variable. Additionally check that the rebuild method works on the output. pa 12 Check that the split method data. Check that the split method also split 'dx' and 'dy'. Check that the split method also split 'dx' and 'dy'. Check that the split method also split 'dx' and 'dy'. pa 13 Check that the split method interprets a negative end interval as a indicates count from end and a zero for the and interval as the end of the vector. Check that the split method also split 'dx' and 'dy'. pa 14 Check that the split method interval as the end of the vector. Check that the split method interval as the end of the vector. Check that the output 2) Check that the 'rebuild' method produces the same object as 'out'. pa 14 Check that the split method interval as the end of the vector. Check that the output have the same 'fs', 'onbw', 'navs', 'xunits'			1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
Interval as a indicates count from end and a zero for the and interval as the end of the vector.1) Check the outputd what the 'rebuild' method pro- duces the same object as 'out'.pa11Check that the [times/frequ] method pass back the output objects to a list of output variables or to a single variable.1) Check the method with a list of output variables and with a sin- gle output variable. Additionally check that the rebuild method works on the output.pa12Check that the split method works with notequal sampled data.Check that the split method works with notequal sampled data.Check that the split method works with notequal sampled data.Check that the split method also split 'dx' and 'dy'.Check that the split method also split 'dx' and 'dy'.Check that the split method also split 'dx' and 'dy'.Check that the split method also split 'dx' and 'dy'.pa14Check that the split method interprets a negative end interval as the end of the vector.Check that the split method also split 'dx' and 'dy'.1) Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'.pa14Check that the split method interval as the end of the vector.1) Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'.pa14Check that the split method interval as the end of the vector.1) Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'.pa14Check that the split method interval as the end of the vector.1) Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'ou	10	Check that the split method interprets a negative end	Check this behavior for time- series data.	pass
11 Check that the [times/frequ] method pass back the output objects to a list of output variables or to a single variable. Call the method with a list of output variables and with a sin- gle output variable. Additionally check that the rebuild method works on the output. pa 12 Check that the split method works with notequal sampled data. Check that the split method works with notequal sampled data. Check that the split method works with notequal sampled data. I) Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'. pa 13 Check that the split method interprets a negative end interval as the end of the vector. Check that the split method interval as the end of the vector. Check that the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'. pa 14 Check that the split method interval as the end of the vector. I) Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'. pa 14 Check that the split method interval as the end of the vector. I) Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'. 3) Check that the output have the same 'fs', 'enbw', 'navs', 'xunits' pa		from end and a zero for the and interval as the end of the vector.	1) Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'.	pass
12Check that the split method works with notequal sampled data.Check that the split method data.Check that the split method data.Check that the split method data.13Check that the split method also split 'dx' and 'dy'.Check that the split method also split 'dx' and 'dy'.Check that the split method also split 'dx' and 'dy'.Deck that the split method also split 'dx' and 'dy' if this values have the same length of the y- values.pa14Check that the split method interprets a negative end interval as a indicates count from end and a zero for the and interval as the end of the vector.Check that the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'.14Check that the split method interval as the end of the vector.Check that the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'.	11	Check that the [times/frequ] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
12 Check that the split method works with notequal sampled data. Check that the split method data. Check that the split method data. works with notequal sampled data. 13 Check that the split method also split 'dx' and 'dy'. Check that the split method also split 'dx' and 'dy'. Check that the split method also split 'dx' and 'dy'. Check that the split method also split 'dx' and 'dy'. Check that the split method also split 'dx' and 'dy'. Check that the split method also split 'dx' and 'dy'. Check that the split method also split 'dx' and 'dy'. Check that the split method also split 'dx' and 'dy'. Check that the split method also split 'dx' and 'dy'. The split 'dx' and 'dy'. 14 Check that the split method interprets a negative end interval as a indicates count from end and a zero for the and interval as the end of the vector. Check that the 'rebuild' method pro- duces the same object as 'out'. 3) Check that the output 2) Check pa that the 'rebuild' method pro- duces the same object as 'out'. 3)		Check that the calit method	tains the same plotinfo plist Check that the split method	neg
131) Check the output 2) Check pa that the 'rebuild' method pro- duces the same object as 'out'.13Check that the split method also split 'dx' and 'dy'.Check that the split method also split 'dx' and 'dy' if this values have the same length of the y- values.pa14Check that the split method interprets a negative end interval as a indicates count from end and a zero for the and interval as the end of the vector.Check that the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'.14	12	works with notequal sampled data.	works with notequal sampled data.	pass
13Check that the split method also split 'dx' and 'dy'.Check that the split method also split 'dx' and 'dy' if this values have the same length of the y- values.pa14Check that the split method interval as a indicates count from end and a zero for the and interval as the end of the vector.Check that the split method interval as the end of the vector.Check that the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'.14Check that the split method interval as a indicates count from end and a zero for the and interval as the end of the vector.Check that the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'. 3) Check that the output have the same 'fs', 'enbw', 'navs', 'xunits'			1) Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'.	pass
141) Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'.pa14Check that the split method interprets a negative end interval as a indicates count from end and a zero for the and interval as the end of the vector.Check this behavior for frequency-series data.pa14Check that the split method interval as a indicates count from end and a zero for the and interval as the end of the vector.Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'. 3) Check that the output have the same 'fs', 'enbw', 'navs', 'xunits'	13	Check that the split method also split 'dx' and 'dy'.	Check that the split method also split 'dx' and 'dy' if this values have the same length of the y- values.	pass
14Check that the split method interprets a negative end interval as a indicates count from end and a zero for the and interval as the end of the vector.Check this behavior for frequency-series data.pa1)Check this behavior for frequency-series data.pa1)Check the output 2)Check pa10Check the output 2)Check pa11Check the output 2)Check pa12Check the output 3)Check that the 'rebuild' method pro- duces the same object as 'out'. 3) Check that the output have the same 'fs', 'enbw', 'navs', 'xunits'			1) Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'.	pass
interval as a indicates count from end and a zero for the and interval as the end of the vector.	14	Check that the split method interprets a negative end	Check this behavior for frequency-series data.	pass
and 'vunits'		interval as a indicates count from end and a zero for the and interval as the end of the vector.	1) Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'. 3) Check that the output have the same 'fs', 'enbw', 'navs', 'xunits' and 'vunits'	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

times/frequ			
15	Check that the split method works with notequal sampled	Check that the split method works with notequal sampled	pass
	data. Check that the interval	data. Check that the interval can	
	can be passed via an array of	be passed via an array of dou-	
	double. Check that the split	ble. Check that the split method	
	method works with 'timeshift'	works with 'timeshift' option.	
	option.	1) Check the outputs 2) Check	pass
		that the 'rebuild' method pro-	
		duces the same object as 'out'.	
16	Check that the split method	Check that the split method	pass
10	works with time intervals which	works with time intervals which	
	doesn' match the sample	doesn' match the sample fre-	
	frequency.	quency.	
		1) Check the outputs 2) Check	pass
		that the 'rebuild' method pro-	
		duces the same object as 'out'.	

Table 132: Unit tests for times/frequ.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



_ao/sqrt			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists	pass
		nave the correct parameters.	
02	works with a vector of objects as input.	works for a vector of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in	pass
		'vec' 2) Check that each output object contains the correct data.	
03	works with a matrix of objects as input.	vorks for a matrix of objects as input.	pass
	, rest	1) Check that the number of el- ements in 'out' is the same as in 'met' 2) Check that each cutruit	pass
		mat ²) Check that each output	
	Tagta that the [ac/aget] method	Test that the [ac/gent] method	pagg
04	works with a list of objects as input.	works for a list of objects as in-	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output	pass
05	Tests that the [ao/sqrt] method works with a mix of different	Tests that the [ao/sqrt] method works with a mix of different ar-	pass
	arrays of objects as input.	 1) Check that the number of elements in 'out' is the same as in ('met' 2) Check that each extract 	pass
		object contains the correct data.	
06	Tests that the [ao/sqrt] method properly applies history.	Test that the result of applying the [ao/sqrt] method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[ao/sqrt]'. 2) Check that the re-built object is the same object as the input.	pass
07	Tests that the [ao/sqrt] method can modify the input AO.	Test that the [ao/sqrt] method can modify the input object by calling with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/sqrt			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the [ao/sqrt]	
		value of the copy 4) Check that	
		out and amodi are the same	
	Test that the [ao/sqrt] method	Test that the [ao/sqrt] method	pass
08	uses the plist to get the axis.	uses the plist to get the axis.	1
		1) Check that the [ao/sqrt]	pass
		method applies to the x-axis 2	1
		Check that the [ao/sqrt] method	
		applies to the y-axis 3) Check	
		that the [ao/sqrt] method applies	
		to both axes 4) Check that the	
		re-built object is the same as in	
		'out[13]'.	
	Test the shape of the data in	Test that the [ao/sqrt] method	pass
09	AOs.	keeps the data shape of the in-	1
		put object. The input AO data	
		must be an array with row data	
		and/or column data.	
		1) Check that the shape of the	pass
		data doesn't change.	P
10	Check that the [ao/sqrt] method	Call the method with a list of	pass
10	pass back the output objects to	output variables and with a sin-	1
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	1
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/sqrt] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	1
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	1

Table 133: Unit tests for a o/sqrt.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/std			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists	pass
		have the correct parameters.	
02	Tests that the [ao/std] method works with a vector of objects as input.	Test that the [ao/std] method works for a vector of objects as input.	pass
		1) Check that the number of elements in 'out' is the same as in	pass
		'vec' 2) Check that each output object contains the correct data.	
03	Tests that the [ao/std] method works with a matrix of objects	Test that the [ao/std] method works for a matrix of objects as	pass
	as input.	1) Check that the number of el- ements in 'out' is the same as in	pass
		'mat' 2) Check that each output object contains the correct data.	
04	Tests that the [ao/std] method works with a list of objects as input.	Test that the [ao/std] method works for a list of objects as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data	pass
05	Tests that the [ao/std] method works with a mix of different arrays of objects as input.	Tests that the [ao/std] method works with a mix of different ar- rays of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
06	Tests that the [ao/std] method properly applies history.	Test that the result of applying the [ao/std] method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[ao/std]'. 2) Check that the re-built object is the same object as the input.	pass
07	Tests that the [ao/std] method can modify the input AO.	Test that the [ao/std] method can modify the input object by calling with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/std			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	-
		is not changed 3) Check that	
		the modified input is the [ao/std]	
		value of the copy 4) Check that	
		out and amodi are the same	
100	Test that the [ao/std] method	Test that the [ao/std] method	pass
108	uses the plist to get the axis.	uses the plist to get the axis.	-
	This is intended to test methods	1) Check that the [ao/std]	pass
	like ao/mean and ao/std which	method applies to the x-axis 2)	1
	return different data types	Check that the [ao/std] method	
	depending on which axis is	applies to the y-axis 3) Check	
	selected.	that the [ao/std] method applies	
		to both axes 4) Check that the	
		re-built object is the same as in	
		'out[13]'.	
00	Test the shape of the data in	Test that the [ao/std] method	pass
09	AOs.	keeps the data shape of the in-	
		put object. The input AO data	
		must be an array with row data	
		and/or column data.	
		1) Check that the shape of the	pass
		data doesn't change.	
10	Check that the [ao/std] method	Call the method with a list of	pass
10	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/std] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 134: Unit tests for ao/std.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/string			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the string method works with a vector of AO objects as input.	Test that the string method works for a vector of AO objects as input.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
03	Tests that the string method works with a matrix of AO objects as input.	Test that the string method works for a matrix of AO objects as input.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
04	Tests that the string method works with a list of AO objects as input.	Test that the string method works for a list of AO objects as input.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
05	Tests that the string method works with a mix of different shaped AO objects as input.	Test that the string method works with an input of matrices and vectors and single AO ob- jects.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
06	Tests that the string method properly applies history.	The method string doesn't change the data, thus it is not possible to check the history. Nothing to do.	pass
07	Tests that the string method doesn't work if the AO object have more than one history step.	The method string throws an er- ror because the input object have more than one history step.	pass
		1	1 10055

Table 135: Unit tests for ao/string.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/sum			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
-	for this method.	for no sets, all sets, and each set	
		1) Check that got Info call no	naga
		1) Check that gettino can re-	pass
		cases 2) Check that all plicts	
		base the correct parameters	
	Tests that the [ap /sum] method	Tagt that the loo gum method	pagg
02	works with a vector of objects	works for a water of objects as	pass
	works with a vector of objects	input	
	as input.	1) Check that the number of el	nase
		oments in 'out' is the same as in	pass
		'vog' 2) Chock that each output	
		object contains the correct data	
	Tests that the [ao/sum] method	Test that the [ao/sum] method	nass
03	works with a matrix of objects	works for a matrix of objects as	pass
	as input	input	
	as input.	1) Check that the number of el-	pass
		ements in 'out' is the same as in	Pabb
		'mat' 2) Check that each output	
		object contains the correct data.	
	Tests that the [ao/sum] method	Test that the [ao/sum] method	pass
04	works with a list of objects as	works for a list of objects as in-	I
	input.	put.	
	1	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/sum] method	Tests that the [ao/sum] method	pass
00	works with a mix of different	works with a mix of different ar-	
	arrays of objects as input.	rays of objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
06	Tests that the [ao/sum] method	Test that the result of applying	pass
	properly applies history.	the [ao/sum] method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[ao/sum]'. 2) Check that the	
		re-built object is the same object	
		as the input.	
07	Tests that the [ao/sum] method	I lest that the [ao/sum] method	pass
	can modify the input AO.	can modify the input object by	
		calling with no output and that	
		input of the function met ti	
		input of the function notation	
		(with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/sum			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the [ao/sum]	
		value of the copy 4) Check that	
		out and amodi are the same	
	Test that the [ao/sum] method	Test that the [ao/sum] method	pass
108	uses the plist to get the axis.	uses the plist to get the axis.	1
	This is intended to test methods	1) Check that the [ao/sum]	pass
	like ao/mean and ao/std which	method applies to the x-axis 2)	Pass
	return different data types	Check that the [ao/sum] method	
	depending on which axis is	applies to the varie 3) Check	
	selected	that the [ao/sum] method an-	
	Scietted.	plies to both aves 4) Check that	
		the re built object is the same as	
		in 'out[1, 2]'	
	Test the shape of the data in	Test that the [as /gum] method	naga
09	AO ₂	less that the [ao/sum] method	pass
	AOS.	keeps the data shape of the m-	
		put object. The input AO data	
		must be an array with row data	
		and/or column data.	
		1) Check that the shape of the	pass
		data doesn't change.	
10	Check that the [ao/sum]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/sum]	Call the method with a list of	pass
11	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 136: Unit tests for a o/sum.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/sumjoin			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the sumjoin method works with a vector of AOs as input.	Test that the sumjoin method works for a vector of AOs as in- put.	pass
		1) Check that the output is exact one AO 2) Check that the output have the correct data. 3) Check the rebuilt object	pass
03	Tests that the sumjoin method works with a matrix of AOs as input.	Tests that the sumjoin method works with a matrix of AOs as input.	pass
		1) Check that the output is exact one AO 2) Check that the output have the correct data.	pass
04	Tests that the sumjoin method works with a list of AOs as input.	Tests that the sumjoin method works with a list of AOs as in- put.	pass
		1) Check that the output is exact one AO 2) Check that the output have the correct data.	pass
05	Tests that the sumjoin method works with a mix of different shaped AOs as input.	Tests that the sumjoin method works with a mix of different shaped AOs as input.	pass
		1) Check that the output is exact one AO 2) Check that the output have the correct data.	pass
06	Tests that the sumjoin method properly applies history.	Test that the result of applying the sumjoin method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'sumjoin'. 2) Check that the re-built object is the same object as the input.	pass
07	Tests that the sumjoin method can modify the input AO.	Test that the sumjoin method can modify the input AO by call- ing with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/sumjoin			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input have the new data	
		4) Check that out and amodi are	
		the same	
08	Test the shape of the output.	Test that the sumjoin method	pass
00		keeps the data shape of the in-	
		put object. The input AO must	
		be an AO with row data and an	
		AO with column data.	
		1) Check that the shape of the	pass
		data doesn't change.	

Table 137: Unit tests for ao/sumjoin.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/svd			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists	pass
		The set of	
02	rests that the syd method	for a wasten of AOa ag input	pass
	works with a vector of AOs as	1) Check that the number of al	
	mput.	1) Check that the humber of el-	pass
		'atroa' 2) Check that each output	
		AQ contains the correct data	
	Tosts that the syd method	AO contains the correct data.	paga
03	works with a matrix of AOs as	for a matrix of AOg ag input	pass
	works with a matrix of AOs as	1) Check that the number of al	naga
	mput.	oments in 'out' is the same as	pass
		in 'atmat' 2) Check that each	
		$\Delta \Omega$ contains the correct	
		data	
	Tests that the syd method	Test that the syd method works	nass
04	works with a list of AOs as	for a list of AOs as input	pass
	input	1) Check that the number of el-	nass
	mpau.	ements in 'out' is the same as in	pass
		input. 2) Check that each output	
		AO contains the correct data.	
	Tests that the syd method	Test that the syd method works	pass
05	works with a mix of different	with an input of matrices and	P
	shaped AOs as input.	vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	1
		input. 2) Check that each output	
		AO contains the correct data.	
06	Tests that the svd method	Test that the result of applying	pass
00	properly applies history.	the svd method can be processed	
		back to an m-file.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'svd'. 2) Check that the re-	
		built object is the same object as	
		'out'.	
07	Tests that the svd method can	Test that the svd method can	pass
~.	modify the input AO.	modify the input AO by call-	
		ing with no output and that the	
		method doesn't change the input	
		of the function notation (with a	
		equal sign).	
		1) Check that 'at4' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		\mid is svd(at4).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/svd			
00		Test that the svd method can	pass
08	Control the method with a plist.	modify the single axis controlled	
		by the plist and the resuld can be	
		processed back to an m-file.	
		1) Check that the svd method	pass
		applies with different options. 2)	
		Check that the re-built objects	
		are the same object as ' $out[12]$ '.	
00		Test that the svd method keeps	pass
09	Control the method with a plist.	the data shape of the input ob-	
		ject. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
10	Check that the svd method pass	Call the method with a list of	pass
10	back the output objects to a list	output variables and with a sin-	
	of output variables or to a single	gle output variable. Additionaly	
	variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 138: Unit tests for ao/svd.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/t0			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the t0 method works with a vector of AOs as input.	The t0 method doesn't work with a vector of AOs. Nothing to do	pass
03	Tests that the t0 method works with a matrix of AOs as input.	The t0 method doesn't work with a matrix of AOs. Nothing to do	pass pass
			pass
04	Tests that the t0 method works with a list of AOs as input.	The t0 method doesn't work with a list of AOs. Nothing to do	pass
			pass
05	Tests that the t0 method works with a mix of different shaped AOs as input.	The t0 method can only return the t0 value of one AO. Nothing to do	pass
			pass
06	Tests that the t0 method properly applies history.	The t0 method doesn't change the AO, thus will no history added. Nothing to do	pass
			pass
07	Tests that the t0 method works for AOs with different data objects.	Test that the t0 method returns the t0 value for AOs with cdata, fsdata, tsdata and xydata ob- jects. 1) Check the output.	pass

Table 139: Unit tests for ao/t0.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/tan			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists	pass
		have the correct parameters.	
02	Tests that the [ao/tan] method works with a vector of objects as input.	Test that the [ao/tan] method works for a vector of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'vec' 2) Check that each output object contains the correct data.	pass
03	Tests that the [ao/tan] method works with a matrix of objects as input.	Test that the [ao/tan] method works for a matrix of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
04	Tests that the [ao/tan] method works with a list of objects as input.	Test that the [ao/tan] method works for a list of objects as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
05	Tests that the [ao/tan] method works with a mix of different arrays of objects as input.	Tests that the [ao/tan] method works with a mix of different ar- rays of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
06	Tests that the [ao/tan] method properly applies history.	Test that the result of applying the [ao/tan] method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[ao/tan]'. 2) Check that the re-built object is the same object as the input.	pass
07	Tests that the [ao/tan] method can modify the input AO.	Test that the [ao/tan] method can modify the input object by calling with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/tan			
/		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	-
		is not changed 3) Check that	
		the modified input is the [ao/tan]	
		value of the copy 4) Check that	
		out and amodi are the same	
	Test that the [ao/tan] method	Test that the [ao/tan] method	pass
08	uses the plist to get the axis.	uses the plist to get the axis.	T
		1) Check that the [ao/tan]	pass
		method applies to the x-axis 2)	P
		Check that the [ao/tan] method	
		applies to the y-axis 3) Check	
		that the $[ao/tan]$ method applies	
		to both axes 4) Check that the	
		re-built object is the same as in	
		out[13]'.	
	Test the shape of the data in	Test that the [ao/tan] method	pass
09	AOs.	keeps the data shape of the in-	1
		put object. The input AO data	
		must be an array with row data	
		and/or column data.	
		1) Check that the shape of the	pass
		data doesn't change.	1
10	Check that the [ao/tan] method	Call the method with a list of	pass
10	pass back the output objects to	output variables and with a sin-	-
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	-
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/tan] method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 140: Unit tests for a o/tan.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/tfe			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the tfe method works with a vector of AOs as input.	Test that the tfe method works for a vector of AOs as input.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
03	Tests that the tfe method doesn't work with a matrix of AOs as input.	Test that the tfe method doesn't work for a matrix of AOs as in- put.	pass
		1) Nothing to check	pass
04	with a list of AOs as input.	for a list of AOs as input.	pass
		1) Check that the number of ele- ments in 'out' is the square of the number in the input. 2) Check that each output AO contains the correct data.	pass
05	Tests that the tfe method doesn't work with a mix of different shaped AOs as input.	Test that the tfe method doesn't work with an input of matrices and vectors and single AOs.	pass
06	Tests that the tfe method properly applies history.	Test that the result of applying the tfe method can be processed back to an m-file.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'tfe'. 2) Check that the re-built object is the same object as 'out'.	pass
07	Tests that the tfe method can not modify the input AO.	Test that the tfe method can not modify the input AO. The method must throw an error for the modifier call.	pass
08	Test the shape of the output.	1) Nothing to check. Test that the plus method keeps the data shape of the input ob- ject. The input AO must be an AO with row data and an AO with column data.	pass pass
		1) Check that the shpe of the output data doesn't change.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/tfe			
	Check that the tfe method pass	This test is not longer necessary	pass
09	back the output objects to a list	because the tfe method pass back	-
	of output variables or to a single	always only one object.	
	variable.	1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
10	Tests that the tfe method agrees	Test that applying the works on	pass
10	with MATLAB's tfestimate	two AOs.	
	when configured to use the same	1) Check that output agrees with	pass
	parameters.	the output of MATLAB's tfesti-	
		mate.	
11	Check that the [ao/tfe] method	Call the method with a list of	pass
11	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	
12	Tests that differently sized data	Test that applying the works on	pass
	sets are treated properly	two AOs.	
		1) Check that the used the length	pass
		of the shortest ao.	
17	Tests handling of units: 1) white	1) Prepare the test tsdata: white	pass
	noise produced from normal	noise from normal distribution +	
	pdf, with a given mean value	offset 2) Assign a random unit	
	and sigma (distribution's 1st	3) Prepare the test tsdata: white	
	and 2nd orders) 2) white noise	noise from normal distribution +	
	produced from normal pdf, with	offset 4) Assign a random unit 5)	
	a given mean value and sigma	tie of the white hoise	
	(distribution's 1st and 2nd	1) Check that (calculated the yu-	pass
	(Torders) 5) the of the white hoise	1) Drepana the test tedates white	naga
21	there is an a far that the set of the	1) Prepare the test tsdata: white	pass
	sotting the Nfft, 1) white poige	offset 2) the of the 2 genies, with	
	produced from normal pdf	out detrending random window	
	with: a given mean value and	set number of averages	
	sigma (distribution's 1st and	1) Check that calculated nave are	nass
	2nd order) 2) the of the 2 series	identical to those requested	pass
	Without beleta willing to the 2 beries,	1) white noise produced from	nass
22	window. set wirdes reflevethers	uniform pdf with a given mean	pass
	Batcheck the Neffect ive hitembers of	value and sigma (distribution's	
	averages] from uniform pdf	1st and 2nd order) 2) the of the	
	with: a given mean value and	time series, without detrending.	
	sigma (distribution's 1st and	random window. random navs 3)	
	2nd order) 2) the of the time	get the number of averages 4) get	
	series, without detrending.	the nfft used 5) run tfe again	
	random window. random navs	with the nfft used	
1	3) get the number of averages 4)		1
	get the nfft used 5) run tfe		
	again, with the nfft used 6)		
	compare the calculated objects		



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/tfe			
		1) Check that calculated objects	pass
		T1 and T2 are identical T_{1}	
0.2	Tests the possibility to set the	1) white noise produced from	pass
20	number of averages rather than	normal pdf, with: a given mean	
	setting the Nfft: 1) white noise	value and sigma (distribution's	
	produced from normal pdf,	1st and 2nd order) 2) the of the	
	with: a given mean value and	time series, without detrending,	
	sigma (distribution's 1st and	random window, random navs 3)	
	2nd order) 2) the of the time	get the number of averages 4) get	
	series, without detrending,	the nfft used 5) run tfe again,	
	random window, random navs	with the nfft used 6) run tfe	
	3) get the number of averages 4)	again, with conflicting parame-	
	get the nfft used 5) run tfe	ters, and verify it uses nfft rather	
	again, with the $nfft$ used 6)	than navs	
	compare navs, nfft, tfes	1) Check that calculated objects	pass
		T1 and T2 are identical 2) Check	
		that T3 used different values	
25	Tests handling of units: 1) white	1) Prepare the test tsdata: white	pass
20	noise produced from normal	noise from normal distribution +	
	pdf, with a given mean value	offset 2) Assign a random unit	
	and sigma (distribution's 1st	3) Prepare the test tsdata: white	
	and 2nd orders) 2) white noise	noise from normal distribution +	
	produced from normal pdf, with	offset 4) Assign a random unit 5)	
	a given mean value and sigma	tfe of the white noise	
	(distribution's 1st and 2nd	1) Check that (calculated tfe yu-	pass
	orders) 3) the of the white noise	nits) equals $[1/Hz]$	
30	Sæsts Hàndhngao£speciahitæsef:	1) Prepare the test tsdata: white	pass
30	theviliptetnesiste putplutced from	noise from normal distribution +	
	normal pdf, with a given mean	offset 2) Assign a random unit 3)	
	value and sigma (distribution's	Prepare the test tsdata: the same	
	1st and 2nd orders) 2) the same	data as 1) and 2) 4) the of the	
	noise series 3) the of the white	series	
	noise series 4) compares the	1) Check that calculated the	pass
	output to unity	equals 1	

Table 141: Unit tests for ao/tfe.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
rule1 (tsdata and tsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and fsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/times			
rule1 (cdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (tsdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (tsdata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (cdata and tsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule1 (xydata and	Tests the arithmetic operators	Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
tsdata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (fsdata and xydata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule1 (cdata and	Tests the arithmetic operators	Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
fsdata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and fsdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule1 (xydata and cdata)	Tests the arithmetic operators rule 1.	Tests the arithmetic operators rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule1 (cdata and	Tests the arithmetic operators	Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
xydata)	rule 1.	rule 1 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	-
		Here we test element-wise opera- tor rule1 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single tsdata and vector tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector tsdata and single tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
mule2 (single	Tasts the exithmatic encretors	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata and vector tsdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector tsdata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single cdata and vector tsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule2 (vector tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and single cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single fsdata and vector fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector fsdata and single fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule2 (single	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata and vector fsdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	1
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector fsdata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single cdata and vector fsdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule2 (vector fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and single cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single xydata and vector xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector xydata and single xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (single cdata and vector xydata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector xydata and single cdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule2 (vector cdata and single cdata)	Tests the arithmetic operators rule 2.	Tests the arithmetic operators rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule2 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 2.	rule 2 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule2 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule ³ (vector tsdata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector tsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector tsdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
nulo2 (vostor edoto	Tasts the exithmatic encretors	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector tsdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule3 (vector fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector xydata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector fsdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector fsdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule3 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector fsdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	F
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector xydata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rulo3 (voctor	Tacts the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
xydata and vector cdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector cdata and vector xydata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule3 (vector cdata and vector cdata)	Tests the arithmetic operators rule 3.	Tests the arithmetic operators rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule3 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 3.	rule 3 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule3 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector tsdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule4 (vector tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector xydata)	rule 4.	rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector tsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector fsdata and vector fsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule4 (vector fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector fsdata)	rule 4.	rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector fsdata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector fsdata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector xydata and vector xydata)	rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector xydata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule4 (vector cdata and vector xydata)	Tests the arithmetic operators rule 4.	Tests the arithmetic operators rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule4 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 4.	rule 4 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	-
		Here we test element-wise opera- tor rule4 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP tsdata and single tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single tsdata and NxP tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule5 (NyP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and single xydata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP tsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP tsdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule5 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP tsdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP fsdata and single fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single fsdata and NxP fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule5 (NxP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and single xydata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP fsdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP fsdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule5 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP fsdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	-
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP xydata and single xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single xydata and NxP xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule5 (NxP xvdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and single cdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	-
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (single cdata and NxP xydata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule5 (NxP cdata and single cdata)	Tests the arithmetic operators rule 5.	Tests the arithmetic operators rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule5 (single cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP cdata)	rule 5.	rule 5 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule5 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule6 (NxP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector xydata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
mula6 (uppedan adata	Tosts the arithmetic encretors	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule6 (NxP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	равя
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule6 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP fsdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	I ma
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule6 (NxP xvdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule6 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 6.	Tests the arithmetic operators rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule6 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP cdata)	rule 6.	rule 6 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule6 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule7 (NxP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule7 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	p 0.00
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule7 (NyP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	равы
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule7 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP fsdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule7 (NxP xydata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and vector cdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule7 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 7.	Tests the arithmetic operators rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule7 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP cdata)	rule 7.	rule 7 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule7 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP tsdata and vector tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector tsdata and NxP tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule8 (NxP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP tsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP tsdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule8 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP tsdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP fsdata and vector fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector fsdata and NxP fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector xydata)	rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP fsdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP fsdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule8 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	Pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP xydata and vector xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector xydata and NxP xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule8 (NyP yvdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and vector cdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (vector cdata and NxP xydata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule8 (NxP cdata and vector cdata)	Tests the arithmetic operators rule 8.	Tests the arithmetic operators rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule8 (vector cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP cdata)	rule 8.	rule 8 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	r
		Here we test element-wise opera- tor rule8 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP tsdata and NxQ tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ tsdata and NxP tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule9 (NyP tsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxQ xydata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP tsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP tsdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule9 (NxQ cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	F
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP fsdata and NxQ fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ fsdata and NxP fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rulo0 (NyP fedata	Tosts the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxQ xydata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP fsdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP fsdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rulo0 (NrO edata	Tosts the arithmetic energy	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP xydata and NxQ xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ xydata and NxP xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule9 (NxP xydata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxQ cdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxQ cdata and NxP xydata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule9 (NxP cdata and NxQ cdata)	Tests the arithmetic operators rule 9.	Tests the arithmetic operators rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule9 (NxQ cdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP cdata)	rule 9.	rule 9 for each data type: xydata, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule9 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP tsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP tsdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
mile10 (NaD edate	Trate the orithmetic encreters	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP tsdata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
rule10 (NxP fsdata	Tests the arithmetic operators	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
and NxP xydata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	-
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP fsdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP fsdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
wile10 (NyP edote	Tests the arithmetic encountered	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
and NxP fsdata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP xydata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/times			
rule10 (NxP	Tests the arithmetic operators	Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built. Tests the arithmetic operators	pass
xydata and NxP cdata)	rule 10.	rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	-
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP xydata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times			
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
rule10 (NxP cdata and NxP cdata)	Tests the arithmetic operators rule 10.	Tests the arithmetic operators rule 10 for each data type: xy- data, fsdata, tsdata, cdata and useful combinations.	pass
		Here we test element-wise opera- tor rule10 as in S2-AEI-TN-3059. 1) Check the data type of the re- sulting object. 2) Check the re- sulting object contains the cor- rect values. 3) Check the error propagation. 4) Check the units of the output object. 5) Check the resulting object can be re- built.	pass
times			pass
tosta (fadata and	Tests all arithmatic aparations	Tests all arithmetic operations	pass
tsdata)	which are not allowed.	which are not allowed.	pass
		Here we test some negative cases.	pass
tests (tsdata and fsdata)	Tests all arithmetic operations which are not allowed.	Tests all arithmetic operations which are not allowed.	pass
		Here we test some negative cases.	pass
tests (AO no data and tsdata)	Tests all arithmetic operations which are not allowed.	Tests all arithmetic operations which are not allowed.	pass
		Here we test some negative cases.	pass
AO no data)	which are not allowed.	which are not allowed.	pass
		Here we test some negative cases.	pass
tests (different fs in tsdata)	Tests all arithmetic operations which are not allowed.	Tests all arithmetic operations which are not allowed.	pass
(1) 0		Here we test some negative cases.	pass
tests (different x	Tests all arithmetic operations	Tests all arithmetic operations	pass
values in fsdata)	which are not allowed.	which are not allowed. Here we test some negative cases.	pass
tests (negative	Tests all arithmetic operations	Tests all arithmetic operations	pass
test)	which are not allowed.	which are not allowed.	-
		Here we test some negative cases.	pass
tests (negative	Tests all arithmetic operations	lests all arithmetic operations	pass
test)	which are not allowed.	which are not allowed.	
		Here we test some negative cases.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/times

Table 142: Unit tests for ao/times.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/timeshift			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the timeshift method	Test that the timeshift method	pass
02	works with a vector of AOs as	works for a vector of AOs as in-	
	input.	put.	
		1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
0.2	Tests that the timeshift method	Test that the timeshift method	pass
03	works with a matrix of AOs as	works for a matrix of AOs as in-	
	input.	put.	
		1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
0.4	Tests that the timeshift method	Test that the timeshift method	pass
04	works with a list of AOs as	works for a list of AOs as input.	
	input.	1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
05	Tests that the timeshift method	Test that the timeshift method	pass
00	works with a mix of different	works with an input of matrices	
	shaped AOs as input.	and vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
06	Tests that the timeshift method	Test that the result of applying	pass
00	properly applies history.	the timeshift method can be pro-	
		cessed back to an m-file.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'timeshift'. 2) Check that the	
		re-built object is the same object	
		as 'out'.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/timeshift			
07	Tests that the timeshift method can modify the input AO	Test that the timeshift method can modify the input AO by call-	pass
		ing with no output and that the	
		method doesn't change the input	
		of the function notation (with a	
		equal sign)	
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	pass
		is timeshift(at1). 3) Check the al-	
		gorithm	
08	Control the method with a plict	Test that the timeshift method	pass
00	Control the method with a plist.	keeps the data shape of the in-	
		put object. The input AO must	
		be an AO with row data and an	
		AO with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
09	Check that the timeshift	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/timeshift]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 143: Unit tests for ao/timeshift.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/transpose			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the [ao/transpose]	Test that the [ao/transpose]	pass
02	method works with a vector of	method works for a vector of ob-	
	objects as input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'vec' 2) Check that each output	
		object contains the correct data.	
03	Tests that the [ao/transpose]	Test that the [ao/transpose]	pass
	method works with a matrix of	method works for a matrix of ob-	
	objects as input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
04	Tests that the [ao/transpose]	Test that the [ao/transpose]	pass
	method works with a list of	method works for a list of objects	
	objects as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/transpose]	Tests that the [ao/transpose]	pass
	method works with a mix of	method works with a mix of dif-	
	different arrays of objects as	ferent arrays of objects as input.	
	input.	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		mat ²) Cneck that each output	
	Tests that the [//	The trans the correct data.	
06	lests that the [ao/transpose]	lest that the result of applying	pass
	method property applies history.	the [ao/transpose] method can	
		1) Check that the last entry in	
		1) Check that the last entry in the history of 'out' comparende	pass
		the mistory of out corresponds	
		the re built object is the same	
		object as the input	
	Tosts that the transpose method	Tost that the transpose method	negg
07	can modify the input AO	can modify the input AO by call	pass
	can mouny the input AO.	ing with no output and that the	
		method doesn't change the input	
		of the function notation (with a	
		or the function notation (with a	
		equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/transpose			
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		is transpose(at1).	
00	Control the method with a plict	Test that the abs method keeps	pass
09	Control the method with a plist.	the data shape of the input ob-	
		ject. The input AO must be an	
		AO with row data and an AO	
		with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
10	Check that the [ao/transpose]	Call the method with a list of	pass
10	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/transpose]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 144: Unit tests for ao/transpose.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/type			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the type method works with a vector of AO objects as input.	Test that the type method works for a vector of AO objects as in- put.	pass
02	Tests that the type method	1) Check the rebuilt output.Test that the type method works	pass pass
05	works with a matrix of AO objects as input.	for a matrix of AO objects as in- put.	
		1) Check the rebuilt output.	pass
04	Tests that the type method works with a list of AO objects	Test that the type method works for a list of AO objects as input.	pass
	as input.	1) Check the rebuilt output.	pass
05	Tests that the type method works with a mix of different shaped AO objects as input.	Test that the type method works with an input of matrices and vectors and single AO objects.	pass
		1) Check the rebuilt output.	pass
06	Tests that the type method properly applies history.	The method type doesn't change the data, thus it is not possible to check the history. Nothing to do.	pass
			pass

Table 145: Unit tests for a o/type.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/uminus			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the [ao/uminus]	Test that the [ao/uminus]	pass
02	method works with a vector of	method works for a vector of	
	objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'vec' 2) Check that each output	
		object contains the correct data.	
03	Tests that the [ao/uminus]	Test that the [ao/uminus]	pass
05	method works with a matrix of	method works for a matrix of	
	objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
04	Tests that the [ao/uminus]	Test that the [ao/uminus]	pass
04	method works with a list of	method works for a list of	
	objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/uminus]	Tests that the [ao/uminus]	pass
00	method works with a mix of	method works with a mix of	
	different arrays of objects as	different arrays of objects as	
	input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
	Tests that the [ao/uminus]	Test that the result of applying	pass
	method properly applies history.	the [ao/uminus] method can be	
		processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[ao/uminus]'. 2) Check that	
		the re-built object is the same	
		object as the input.	
07	Tests that the [ao/uminus]	Test that the [ao/uminus]	pass
	method can modify the input	method can modify the input	
	AO.	object by calling with no output	
		and that the method doesn't	
		change the input of the function	
		notation (with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/uminus			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the [ao/uminus]	
		value of the copy 4) Check that	
		out and amodi are the same	
00	Test that the [ao/uminus]	Test that the [ao/uminus]	pass
08	method uses the plist to get the	method uses the plist to get the	
	axis.	axis.	
		1) Check that the [ao/uminus]	pass
		method applies to the x-axis	
		2) Check that the [ao/uminus]	
		method applies to the y-axis	
		3) Check that the [ao/uminus]	
		method applies to both axes 4)	
		Check that the re-built object is	
		the same as in ' $out[13]$ '.	
00	Test the shape of the data in	Test that the [ao/uminus]	pass
09	AOs.	method keeps the data shape of	
		the input object. The input AO	
		data must be an array with row	
		data and/or column data.	
		1) Check that the shape of the	pass
		data doesn't change.	
10	Check that the [ao/uminus]	Call the method with a list of	pass
10	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/uminus]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 146: Unit tests for ao/uminus.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/unwrap			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the unwrap method works with a vector of AOs as input.	Test that the unwrap method works for a vector of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'avec' 2) Check that each ouput corrects the phase angles	pass
03	Tests that the unwrap method works with a matrix of AOs as input.	Tests that the unwrap method works with a matrix of AOs as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'amat' 2) Check that each ouput corrects the phase angles	pass
04	Tests that the unwrap method works with a list of AOs as input.	Tests that the unwrap method works with a list of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in the input 2) Check that each ouput corrects the phase angles	pass
05	Tests that the unwrap method works with a mix of different shaped AOs as input.	Tests that the unwrap method works with a mix of different shaped AOs as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in the input 2) Check that each ouput corrects the phase angles	pass
06	Tests that the unwrap method properly applies history.	Test that the result of applying the unwrap method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'unwrap'. 2) Check that the re-built object is the same object as the input.	pass
07	Tests that the unwrap method can modify the input AO.	Test that the unwrap method can modify the input AO by call- ing with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/unwrap			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the unwraped	
		value of the copy 4) Check that	
		out and amodi are the same	
	Test that the unwrap method	Test that the unwrap method	pass
08	uses the plist to get the axis.	uses the plist to get the axis.	1
		1) Check that the unwrap	pass
		method applies to the x-axis 2	1
		Check that the unwrap method	
		applies to the v-axis 3) Check	
		that the unwrap method applies	
		to both axes 4) Check that the	
		re-built object is the same as in	
		'out[13]'.	
		Test that the unwrap method	pass
09	Test the shape of the output.	keeps the data shape of the in-	-
		put object. The input AO must	
		be an AO with row data and an	
		AO with column data.	
		1) Check that the shape of the	pass
		data doesn't change.	-
10	Check that the unwrap method	Call the method with a list of	pass
10	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/unwrap]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 147: Unit tests for ao/unwrap.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/upsample			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the upsample method	Test that the upsample method	pass
02	works with a vector of AOs as	works for a vector of AOs as in-	
	input.	put.	
		1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
02	Tests that the upsample method	Test that the upsample method	pass
03	works with a matrix of AOs as	works for a matrix of AOs as in-	
	input.	put.	
		1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
0.4	Tests that the upsample method	Test that the upsample method	pass
04	works with a list of AOs as	works for a list of AOs as input.	
	input.	1) Check that the number of ele-	pass
		ments in 'out' is the square of the	-
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
05	Tests that the upsample method	Test that the upsample method	pass
05	works with a mix of different	works with an input of matrices	-
	shaped AOs as input.	and vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	-
		input. 2) Check that each output	
		AO contains the correct data.	
00	Tests that the upsample method	Test that the result of applying	pass
06	properly applies history.	the upsample method can be pro-	-
		cessed back to an m-file.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	-
		to 'upsample'. 2) Check that the	
		re-built object is the same object	
		as 'out'.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/upsample			
07	Tests that the upsample method	Test that the upsample method	pass
	can modify the input AO.	can modify the input AO by call-	
		ing with no output and that the	
		method doesn't change the input	
		of the function notation (with a	
		equal sign).	
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		is $upsample(at1)$. 3) Check the	
		algorithm	
		Test that the upsample method	pass
08	Control the method with a plist.	keeps the data shape of the in-	
		put object. The input AO must	
		be an AO with row data and an	
		AO with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	1
	Check that the upsample	Call the method with a list of	pass
09	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/upsample]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	-
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 148: Unit tests for ao/upsample.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/var			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the [ao/var] method	Test that the [ao/var] method	pass
02	works with a vector of objects	works for a vector of objects as	
	as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'vec' 2) Check that each output	
		object contains the correct data.	
03	Tests that the [ao/var] method	Test that the [ao/var] method	pass
03	works with a matrix of objects	works for a matrix of objects as	
	as input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
04	Tests that the [ao/var] method	Test that the [ao/var] method	pass
04	works with a list of objects as	works for a list of objects as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
05	Tests that the [ao/var] method	Tests that the [ao/var] method	pass
00	works with a mix of different	works with a mix of different ar-	
	arrays of objects as input.	rays of objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mat' 2) Check that each output	
		object contains the correct data.	
06	Tests that the [ao/var] method	Test that the result of applying	pass
	properly applies history.	the [ao/var] method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[ao/var]'. 2) Check that the	
		re-built object is the same object	
		as the input.	
07	Tests that the $[ao/var]$ method	Test that the [ao/var] method	pass
	can modify the input AO.	can modify the input object by	
		calling with no output and that	
		the method doesn't change the	
		input of the function notation	
		(with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/var			
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that	
		the modified input is the [ao/var]	
		value of the copy 4) Check that	
		out and amodi are the same	
100	Test that the [ao/var] method	Test that the [ao/var] method	pass
108	uses the plist to get the axis.	uses the plist to get the axis.	-
	This is intended to test methods	1) Check that the [ao/var]	pass
	like ao/mean and ao/std which	method applies to the x-axis 2	1
	return different data types	Check that the [ao/var] method	
	depending on which axis is	applies to the y-axis 3) Check	
	selected.	that the [ao/var] method applies	
		to both axes 4) Check that the	
		re-built object is the same as in	
		out[13]'.	
	Test the shape of the data in	Test that the [ao/var] method	pass
09	AOs	keeps the data shape of the in-	P
		put object. The input AO data	
		must be an array with row data	
		and/or column data	
		1) Check that the shape of the	nass
		data doesn't change	pass
	Check that the [ao/yar] method	Call the method with a list of	nass
10	pass back the output objects to	output variables and with a sin-	pass
	a list of output variables or to a	gle output variable Additionaly	
	single variable	check that the rebuild method	
		works on the output	
		1) Check that the output con-	nass
		tains the right number of ob-	Pass
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
	Check that the [ao/var] method	Call the method with a list of	pass
11	pass back the output objects to	output variables and with a sin-	P abb
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output	
		1) Check that the output con-	pass
		tains the same plotinfo plist	Pass

Table 149: Unit tests for a o/var.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/whiten1D			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	pass
		cases 2) Check that all plists	
		have the correct parameters	
	Tosts that the whiten1D	Tost that the whiten1D method	nace
02	method works with a voctor of	works for a vector of AOs as in	pass
	A Qa ag input	works for a vector of AOS as in-	
	AOs as input.	1) Check that the number of el	pagg
		1) Check that the humber of el-	pass
		'atroc' 2) Check that each autout	
		AC contains the contrast data 2)	
		AO contains the correct data. 5)	
		Check that each output AO con-	
		tains empty yunits	
03	lests that the whitehild	lest that the whitehild method	pass
	method works with a matrix of	works for a matrix of AOs as in-	
	AOs as input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as	
		in 'atmat' 2) Check that each	
		output AO contains the correct	
		data. 3) Cneck that each output	
		AO contains empty yunits	
04	Tests that the whitenID	Test that the whiten1D method	pass
	method works with a list of AOs	works for a list of AOs as input.	
	as input.	1) Check that the number of el-	pass
		ements in out is the same as in	
		input. 2) Cneck that each output	
		AU contains the correct data. 3)	
		Check that each output AO con-	
		tains empty yunits	
05	Tests that the whitehild	Test that the whitehild method	pass
	lifferent shared AOs as insert	works with an input of matrices	
	different snaped AOs as input.	and vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in out is the same as in	
		mput. 2) Check that each output	
		AU contains the correct data. 3)	
		Check that each output AO con-	
		tains empty yunits	
06	lests that the whitenID	Lest that the result of apply-	pass
	method properly applies history.	ing the whitenin method can be	
		processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to whiten1D'. 2) Check that the	
		re-built object is the same object	
		as the input.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/whiten1D			
07	Tests that the whiten1D method	Test that the whiten1D method	pass
	can modify the input AO.	can modify the input AO by call-	
		ing with no output and that the	
		method doesn't change the input	
		of the function notation (with a	
		equal sign).	
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		is whiten1D(at1).	
08	Test the shape of the output	Test that the whiten1D method	pass
		keeps the data shape of the in-	
		put object. The input AO must	
		be an AO with row data and an	
		AO with column data.	
		1) Check that the shape of the	pass
		data doesn't change. 2) Check	
		that the output AOs have empty	
		yunits	
00	Check that the whiten1D	Call the method with a list of	pass
05	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
10	Check that the whiten1D	Generate a fixed series of noise	pass
10	method is capable to increase	data, apply a filter to colour data	
	spectral flatness in case of no	and run the method with a cer-	
	model input	tain number of parameters.	
		1) Check that the output spec-	pass
		trum is flatter than input colored	
		spectrum 2) Check that the out-	
		put AOs have empty yunits	
11	Check that the whiten1D	Generate a fixed series of noise	pass
	method is capable to increase	data, apply a filter to colour data	
	spectral flatness when a model	and run the method with a cer-	
	is input	tain number of parameters.	
		1) Check that the output spec-	pass
		trum is flatter than input colored	
		spectrum	

Table 150: Unit tests for ao/whiten1D.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/whiten2D			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the whiten2D	Test that the whiten2D method	pass
02	method works with a vector of	works for a vector of AOs as in-	
	AOs as input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'acv2' 2) Check that each output	
		AO contains the correct data.	
03	Tests that the whiten2D	Test that the whiten2D method	pass
	method works with a matrix of	works for a matrix of AOs as in-	
	AOs as input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as	
		in 'atmat' 2) Check that each	
		output AO contains the correct	
		data.	
04	Tests that the whiten2D	Test that the whiten2D method	pass
01	method works with a list of AOs	works for a list of AOs as input.	
	as input.	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
05	Tests that the whiten2D	Test that the whiten1D method	pass
	method works with a mix of	works with an input of matrices	
	different shaped AOs as input.	and vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AU contains the correct data.	
06	Tests that the whiten2D	Test that the result of apply-	pass
	method properly applies history.	ing the whiten2D method can be	
		processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to whiten2D'. 2) Check that the	
		re-built object is the same object	
		as the input.	
07	Tests that the whiten2D method	1 Test that the whiten2D method	pass
	can modify the input AO.	can modify the input AO by call-	
		ing with no output and that the	
		method doesn't change the input	
		of the function notation (with a	
		equal sign).	
		1) Nothing to do.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/whiten2D			
08	Test the shape of the output.	Test that the whiten2D method keeps the data shape of the in-	pass
		put object. The input AO must	
		be an AO with row data and an	
		AO with column data.	
		1) Check that the shape of the	pass
		data doesn't change.	
00	Check that the noisegen2D	Call the method with a list of	pass
03	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 151: Unit tests for ao/whiten2D.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/x			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the x method works	The x method doesn't work with	pass
02	with a vector of AOs as input.	a vector of AOs. Nothing to do	
			pass
03	Tests that the x method works	The x method doesn't work with	pass
00	with a matrix of AOs as input.	a matrix of AOs. Nothing to do	
			pass
04	Tests that the x method works	The x method doesn't work with	pass
01	with a list of AOs as input.	a list of AOs. Nothing to do	
			pass
05	Tests that the x method works	The x method can only return	pass
	with a mix of different shaped	the x values of one AO. Nothing	
	AOs as input.	to do	
			pass
06	Tests that the x method	The x method doesn't change the	pass
	properly applies history.	AO, thus will no history added.	
		Nothing to do	
			pass
07	Tests that the x method works	Test that the x method returns	pass
	for AOs with different data	the x values for AOs with cdata,	
	objects.	fsdata, tsdata and xydata ob-	
		jects.	
		1) Check the output.	pass
08	Tests that the x method returns	Test that the x method returns	pass
	the x values of the data object	the x values in a column vector	
		independent form the shape of	
		the values in the data object.	
		1) Check that 'x1' and 'x2' are	pass
		column vectors.	

Table 152: Unit tests for a o/x.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/xcorr			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the xcorr method	Test that the xcorr method	pass
02	works with a vector with 2 AOs	works for a vector with 2 AOs as	
	as input.	input.	
		1) Check that the numbers of	pass
		outputs is 1 2) Check that the	
		output AO contains the correct	
		data. 3) Check that the units are	
		properly handled	
03	Tests that the xcorr method	Tests that the xcorr method does	pass
	does not work with a matrix of	not work with a matrix of AOs as	
	AOs as input.	input.	
		1) Nothing to check	pass
04	Tests that the xcorr method	Tests that the xcorr method	pass
04	works with a list of 2 AOs as	works with a list of 2 AOs as in-	
	input.	put.	
		1) Check that the numbers of	pass
		outputs is 1 2) Check that the	
		output AO contains the correct	
		data. 3) Check that the units are	
		properly handled	
05	Tests that the xcorr method	Tests that the xcorr method does	pass
05	does not work with a mix of	not work with a mix of different	
	different shaped AOs as input.	shaped AOs as input.	
		1) Nothing to check	pass
06	Tests that the xcorr method	Test that the result of applying	pass
00	properly applies history.	the xcorr method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'xcorr'. 2) Check that the re-	
		built object is the same object as	
		the input.	
07	Test the shape of the output	Test that the xcorr method keeps	pass
	rest the shape of the output.	the data shape of the input ob-	
		ject. In this case the first AO de-	
		fines the data shape.	
		1) Check that the shape of the	pass
		data doesn't change.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

ao/xcorr			
ao/xcorr 08	Check that the xcorr method pass back the output objects to a list of output variables or to a single variable. Does not make sense anymore, given the structure double input -> single output	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output. Addition- aly check that the units are han- dled properly. Additionaly check that the symmetry is preserved. 1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob-	pass
09	Check that the xcorr method	ject as 'out'. 3) Check that the units are handled properly. 4) Check that the symmetry is pre- served. Check that the xcorr method re-	pass
	re-sample all input AOs to the highest frequency.	sample all input AOs to the high- est frequency.	nase
		tains the right number of objects 2) Check that the first AO is resampled to '200' 3) Check that the 'rebuild' method produces the same object as 'out'.	pass
10	Check that the xcorr method truncate all input AOs to the same length.	Check that the xcorr method truncate all input AOs to the same length.	pass
		 Check that the output contains the right number of objects Check that the second AO is truncated to 10 seconds 3) Check that the 'rebuild' method produces the same object as 'out'. 	pass
11	Check that the [ao/xcorr] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the same plotinfo plist	pass
12	Check that the xcorr uses different lag ranges [-maxlags:maxlags] Check that the xcorr accepts different scales options Check that the xcorr handles units properly	 Check that the xcorr uses different lag ranges [- maxlags:maxlags] 2) Check that the xcorr accepts different scales options Check that the output have 	pass
	nancies units property	the range [-maxlags:maxlags] 2) Check that the output have correct units	pass


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/xcorr			
19	Check that the xcorr uses	Check that the xcorr uses	pass
10	different lag ranges	different lag ranges [-	
	[-maxlags:maxlags]	maxlags:maxlags]	
		1) Check that the output have	pass
		the range [-maxlags:maxlags]	

Table 153: Unit tests for ao/xcorr.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/xunits			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.9	Tests that the xunits method	Tests that the xunits method	pass
02	works with a vector of AOs as	works with a vector of AOs as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as	
		in the input. 2) Check that each	
		output unit object contains the	
		correct values.	
0.2	Tests that the xunits method	Tests that the xunits method	pass
00	works with a matrix of AOs as	works with a matrix of AOs as	
	input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as	
		in the input. 2) Check that each	
		output unit object contains the	
		correct values.	
04	Tests that the xunits method	Tests that the xunits method	pass
04	works with a list of AOs as	works with a list of AOs as in-	
	input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as	
		in the input. 2) Check that each	
		output unit object contains the	
		correct values.	
05	Tests that the xunits method	Tests that the xunits method	pass
	works with a mix of different	works with a mix of different	
	shaped AOs as input.	shaped AOs as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as	
		in the input. 2) Check that each	
		output unit object contains the	
		correct values.	
06	Iests that the xunits method	I ne xunits method doesn't	pass
	properly applies history.	change the AO, thus will no	
		history added. Nothing to do	
			pass
07	Tests that the xunits method	Test that the xunits method re-	pass
	works for AOs with different	turns the xunits values for AOs	
	data objects.	with isdata, tsdata and xydata	
		with edate	
		WITH COATA.	
		1) Uneck the output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/xunits

Table 154: Unit tests for ao/xunits.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/y			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the y method works	The y method doesn't work with	pass
02	with a vector of AOs as input.	a vector of AOs. Nothing to do	
			pass
03	Tests that the y method works	The y method doesn't work with	pass
	with a matrix of AOs as input.	a matrix of AOs. Nothing to do	
			pass
04	Tests that the y method works	The y method doesn't work with	pass
	with a list of AOs as input.	a list of AOs. Nothing to do	
			pass
05	Tests that the y method works	The y method can only return	pass
	with a mix of different shaped	the y values of one AO. Nothing	
	AOs as input.	to do	
			pass
06	Tests that the y method	The y method doesn't change the	pass
	properly applies history.	AO, thus will no history added.	
		Nothing to do	
			pass
07	Tests that the y method works	Test that the y method returns	pass
	for AOs with different data	the y values for AOs with cdata,	
	objects.	Isdata, tsdata and xydata ob-	
		jects.	
		1) Check the output.	pass
08	Tests that the y method returns	Test that the y method returns	pass
	the y values of the data object	the y values in a column vector	
		independent form the shape of	
		the values in the data object.	
		1) Check that 'y1' and 'y2' are	pass
		column vectors.	

Table 155: Unit tests for a o/y.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/yunits			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the yunits method works with a vector of AOs as input.	Tests that the yunits method works with a vector of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in the input. 2) Check that each output unit object contains the correct values.	pass
03	Tests that the yunits method works with a matrix of AOs as input.	Tests that the yunits method works with a matrix of AOs as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in the input. 2) Check that each output unit object contains the correct values.	pass
04	Tests that the yunits method works with a list of AOs as input.	Tests that the yunits method works with a list of AOs as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in the input. 2) Check that each output unit object contains the correct values.	pass
05	Tests that the yunits method works with a mix of different shaped AOs as input.	Tests that the yunits method works with a mix of different shaped AOs as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in the input. 2) Check that each output unit object contains the correct values.	pass
06	Tests that the yunits method properly applies history.	The yunits method doesn't change the AO, thus will no history added. Nothing to do	pass
07	Tests that the yunits method works for AOs with different data objects.	Test that the yunits method re- turns the yunits values for AOs with cdata, fsdata, tsdata and xydata objects.	pass

Table 156: Unit tests for ao/yunits.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/zDomainFit			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the zDomainFit method works with a vector of AOs as input.	Test that the zDomainFit method works for a vector of AOs as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'av' 2) Check that each output AO contains the correct data.	pass
03	Tests that the zDomainFit method works with a matrix of AOs as input.	Test that the zDomainFit method works for a matrix of AOs as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'am' 2) Check that each output AO contains the correct data.	pass
04	Tests that the zDomainFit method works with a list of AOs as input.	Test that the zDomainFit method works for a list of AOs as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
05	Tests that the zDomainFit method works with a mix of different shaped AOs as input.	Test that the zDomainFit method works with an input of matrices and vectors and single AOs.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output AO contains the correct data.	pass
06	Tests that the zDomainFit method properly applies history.	Test that the result of applying the zDomainFit method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'zDomainFit'. 2) Check that the re-built object is the same object as the input.	pass
07	zDomainFit cannot modify the input AO.	Test that the sDomainFit method can modify the input AO by calling with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/zDomainFit			
		1) Nothing to do.	pass
00	Check that the zDomainFit	Call the method with a list of	pass
09	method pass back the output	output variables and with a sin-	
	objects to a single variable	gle output variable. Additionaly	
	correctly.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
10	Tests that the zDomainFit	Test that the zDomainFit	pass
10	method return the correct	method works with an input of	
	coefficients	matrices and vectors and single	
		AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that output con-	
		tains the correct coefficients.	

Table 157: Unit tests for ao/zDomainFit.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ao/zeropad			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the zeropad method	Test that the zeropad method	pass
02	works with a vector of AOs as	works for a vector of AOs as in-	
	input.	put.	
		1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
0.2	Tests that the zeropad method	Test that the zeropad method	pass
03	works with a matrix of AOs as	works for a matrix of AOs as in-	
	input.	put.	
		1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
0.4	Tests that the zeropad method	Test that the zeropad method	pass
04	works with a list of AOs as	works for a list of AOs as input.	
	input.	1) Check that the number of ele-	pass
		ments in 'out' is the square of the	
		number in the input. 2) Check	
		that each output AO contains	
		the correct data.	
05	Tests that the zeropad method	Test that the zeropad method	pass
00	works with a mix of different	works with an input of matrices	
	shaped AOs as input.	and vectors and single AOs.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		AO contains the correct data.	
06	Tests that the zeropad method	Test that the result of applying	pass
00	properly applies history.	the zeropad method can be pro-	
		cessed back to an m-file.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'zeropad'. 2) Check that the	
		re-built object is the same object	
		as 'out'.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ao/zeropad			
07	Tests that the zeropad method	Test that the zeropad method	pass
07	can modify the input AO.	can modify the input AO by call-	
		ing with no output and that the	
		method doesn't change the input	
		of the function notation (with a	
		equal sign).	
		1) Check that 'at1' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		is zeropad(at1).	
00	Check the data shape of the	Test that the zeropad method	pass
08	output against the input.	keeps the data shape of the in-	
		put object. The input AO must	
		be an AO with row data and an	
		AO with column data.	
		1) Check that the shpe of the	pass
		data doesn't change.	
00	Check that the zeropad method	Call the method with a list of	pass
09	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [ao/zeropad]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 158: Unit tests for ao/zeropad.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



collection/coll	ection		
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually. 1) Check that getInfo call re-	pass
		turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	
02	Tests that the [collection/collection] method works with a vector of objects	Test that the [collec- tion/collection] method works for a vector of objects as input.	pass
	as input.	1) Check that the number of el- ements in 'out' is the same as in 'vec' 2) Check that each output object contains the correct data.	pass
03	Tests that the [collection/collection] method works with a matrix of objects	Test that the [collec- tion/collection] method works for a matrix of objects as input.	pass
	as input.	1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
04	Tests that the [collection/collection] method works with a list of objects as	Test that the [collec- tion/collection] method works for a list of objects as input.	pass
	input.	1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
05	Tests that the [collection/collection] method works with a mix of different arrays of objects as input.	Tests that the [collec- tion/collection] method works with a mix of different arrays of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
06	Tests that the [collection/collection] method properly applies history.	Test that the result of applying the [collection/collection] method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[collection/collection]'. 2) Check that the re-built object is the same object as the input.	pass
07	Tests that the collection method properly applies history to the constructor with different numbers of inputs.	Test that the output can be processed back with the rebuild method.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



collection/co	ollection		
		1) Check that the last entry in the history of 'out' corresponds to 'collection'. 2) Check that the rebuilt objects are the same	pass
08	Tests the collection constructor with different inputs.	Tests the collection constructor with different inputs.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'collection'. 2) Check that the rebuilt objects are the same	pass
60	Tests that the constructor method doesn't apply history to the read MAT-file constructor.	Tests that the constructor method doesn't apply history to the read MAT-file constructor.	pass
		1) Check that the history is the same as the history of the saved object. Because save and load shouldn't add a history step. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
61	Tests that the constructor properly applies history to the read XML-file constructor.	Tests that the constructor prop- erly applies history to the read XML-file constructor.	pass
		1) Check that the history is the same as the history of the saved object. Because save and load shouldn't add a history step. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
62	Tests that the constructor properly applies history in the struct constructor.	Tests that the constructor prop- erly applies history in the struct constructor.	pass
		1) Check that the last entry in the history of 'out' corresponds to the class name. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
64	Tests that the contructor properly applies history to the plist(filename) constructor.	 Tests that the contructor properly applies history to the plist(filename) constructor. 1) Check that the save method doesn't change the input object 2) Check that the save method 	pass
		ject 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	
65	Tests that the contructed object can be submitted and retrieved.	Tests that the contructed object can be submitted and retrieved.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

collection/collection	o n		
		1) Check that the last entry in the history of 'out' corresponds to the class name. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
68	Tests that the constructor properly applies history to the conn+Id constructor.	Tests that the constructor properly applies history to the conn+Id constructor. 1) Check that the last entry in the history of 'out' corresponds to class name. 2) Check that the 'rebuild' method produces the name chiest as 'out'	pass
70	Tests that the constructor properly applies history to the plist(<plist-object>) constructor.</plist-object>	Test that the output can be processed back with the rebuild method. 1) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass

Table 159: Unit tests for collection/collection.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

collection/copy			
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	

Table 160: Unit tests for collection/copy.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



collection/loadobj			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check the shape of the loaded	pass
		objects.	

Table 161: Unit tests for collection/loadobj.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

filterbank/copy			
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	

Table 162: Unit tests for filterbank/copy.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

filterbank/filterba	nk		
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the [filterbank/filterbank] method works with a vector of objects	Test that the [filter- bank/filterbank] method works for a vector of objects as input.	pass
	as input.	1) Check that the number of el- ements in 'out' is the same as in 'vec' 2) Check that each output object contains the correct data.	pass
03	Tests that the [filterbank/filterbank] method works with a matrix of objects	Test that the [filter- bank/filterbank] method works for a matrix of objects as input.	pass
	as input.	1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
04	Tests that the [filterbank/filterbank] method works with a list of objects as	Test that the [filter- bank/filterbank] method works for a list of objects as input.	pass
	input.	1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
05	Tests that the [filterbank/filterbank] method works with a mix of different arrays of objects as input.	Tests that the [filter- bank/filterbank] method works with a mix of different arrays of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
06	Tests that the [filterbank/filterbank] method properly applies history.	Test that the result of ap- plying the [filterbank/filterbank] method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[filterbank/filterbank]'. 2) Check that the re-built object is the same object as the input.	pass
60	Tests that the constructor method doesn't apply history to the read MAT-file constructor.	Tests that the constructor method doesn't apply history to the read MAT-file constructor.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

filterbank/filterba	nk		
		1) Check that the history is the same as the history of the saved object. Because save and load shouldn't add a history step. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
61	Tests that the constructor properly applies history to the read XML-file constructor.	Tests that the constructor prop- erly applies history to the read XML-file constructor.	pass
		1) Check that the history is the same as the history of the saved object. Because save and load shouldn't add a history step. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
62	Tests that the constructor properly applies history in the struct constructor.	Tests that the constructor prop- erly applies history in the struct constructor.	pass
		1) Check that the last entry in the history of 'out' corresponds to the class name. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
64	Tests that the contructor properly applies history to the plist(filename) constructor.	Tests that the contructor prop- erly applies history to the plist(filename) constructor.	pass
		1) Check that the save method doesn't change the input ob- ject 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
65	Tests that the contructed object can be submitted and retrieved.	Tests that the contructed object can be submitted and retrieved.	pass
		the history of 'out' corresponds to the class name. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
68	Tests that the constructor properly applies history to the conn+Id constructor.	Tests that the constructor properly applies history to the conn+Id constructor.	pass
		1) Check that the last entry in the history of 'out' corresponds to class name. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass

Table 163: Unit tests for filterbank/filterbank.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



filterbank/loadobj			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check the shape of the loaded	pass
		objects.	

Table 164: Unit tests for filterbank/loadobj.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

matrix/char			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the char method	Test that the char method works	pass
02	works with a vector of MATRIX	for a vector of MATRIX objects	
	objects as input.	as input.	
		1) Check that the output contain	pass
		at least the character set (char-	
		method) of all inner object	
03	Tests that the char method	Test that the char method works	pass
00	works with a matrix of	for a matrix of MATRIX objects	
	MATRIX objects as input.	as input.	
		1) Check that the output contain	pass
		at least the character set (char-	
		method) of all inner object	
04	Tests that the char method	Test that the char method works	pass
04	works with a list of MATRIX	for a list of MATRIX objects as	
	objects as input.	input.	
		1) Check that the output contain	pass
		at least the character set (char-	
		method) of all inner object	
05	Tests that the char method	Test that the char method works	pass
00	works with a mix of different	with an input of matrices and	
	shaped MATRIX objects as	vectors and single MATRIX ob-	
	input.	jects.	
		1) Check that the output contain	pass
		at least the character set (char-	
		method) of all inner object	
06	Tests that the char method	The method char doesn't change	pass
	properly applies history.	the data, thus it is not possible	
		to check the history. Nothing to	
		do.	
			pass

Table 165: Unit tests for matrix/char.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

matrix/copy			
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	

Table 166: Unit tests for matrix/copy.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



matrix/ctranspose			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the ctranspose method works with a vector of MATRICES as input.	Test that the ctranspose method works for a vector of MATRICES as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mav' 2) Check that each output contains the correct data.	pass
03	Tests that the ctranspose method works with a matrix of MATRICES as input.	Test that the ctranspose method works for a matrix of MATRI- CES as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mam' 2) Check that each output contains the correct data.	pass
04	Tests that the ctranspose method works with a list of MATRICES as input.	Test that the ctranspose method works for a list of MATRICES as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the ctranspose method works with a mix of different shaped MATRICES as input.	Test that the ctranspose method works with an input of matrices and vectors and single MATRI- CES.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the ctranspose method properly applies history.	Test that the result of applying the ctranspose method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'ctranspose'. 2) Check that the re-built object is the same object as 'out'.	pass
07	Tests that the ctranspose method can modify the input MATRIX.	Test that the ctranspose method can modify the input MATRIX by calling with no output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

matrix/ctranspose			
		1) Check that 'out' and 'obj_eq'	pass
		are now different. 2) Check that	
		'obj_eq' is not changed 3) Check	
		that the modified input is the	
		ctranspose value of the copy 4)	
		Check that out and amodi are	
		the same	
08	Tests that the ctranspose	Tests that the ctranspose	pass
00	method accepts plist as an	method accepts plist as an	
	input.	input.	
		1) Check that the correct data 2)	pass
		Check that the re-built object is	
		the same object as 'out'.	
11	Check that the	Call the method with a list of	pass
	[matrix/ctranspose] method	output variables and with a sin-	
	pass back the output objects to	gle output variable. Additionaly	
	a list of output variables or to a	check that the rebuild method	
	single variable.	works on the output.	
	~	1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 167: Unit tests for matrix/ctranspose.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

matrix/display			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set	pass
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the display method	Test that the display method	pass
02	works with a vector of MATRIX	works for a vector of MATRIX	
	objects as input.	objects as input.	
		1) Check that the output contain	pass
		at least each object name	
03	Tests that the display method	Test that the display method	pass
	works with a matrix of	works for a matrix of MATRIX	
	MATRIX objects as input.	objects as input.	
		1) Check that the output contain	pass
		at least each object name	
04	Tests that the display method	Test that the display method	pass
01	works with a list of MATRIX	works for a list of MATRIX ob-	
	objects as input.	jects as input.	
		1) Check that the output contain	pass
		at least each object name	
05	Tests that the display method	Test that the display method	pass
00	works with a mix of different	works with an input of matrices	
	shaped MATRIX objects as	and vectors and single MATRIX	
	input.	objects as.	
		1) Check that the output contain	pass
		at least each object name	
06	Tests that the display method	The method display doesn't	pass
	properly applies history.	change the data, thus it is not	
		possible to check the history.	
		Nothing to do.	
			pass

Table 168: Unit tests for matrix/display.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



matrix/loadobj			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check the shape of the loaded	pass
		objects.	

Table 169: Unit tests for matrix/loadobj.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



matrix/matrix			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists	pass
		have the correct parameters.	
02	Tests that the [matrix/matrix] method works with a vector of	Test that the [matrix/matrix] method works for a vector of ob-	pass
	objects as input.	 Check that the number of elements in 'out' is the same as in 	pass
	Tosts that the [matrix/matrix]	'vec' 2) Check that each output object contains the correct data.	page
03	method works with a matrix of objects as input.	method works for a matrix of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
04	Tests that the [matrix/matrix] method works with a list of objects as input.	Test that the [matrix/matrix] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
05	Tests that the [matrix/matrix] method works with a mix of different arrays of objects as	Tests that the [matrix/matrix] method works with a mix of dif- ferent arrays of objects as input.	pass
	mput.	1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
06	Tests that the [matrix/matrix] method properly applies history.	Test that the result of applying the [matrix/matrix] method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[matrix/matrix]'. 2) Check that the re-built object is the same object as the input.	pass
07	Tests that the matrix method properly applies history to the constructor with different	Test that the output can be processed back with the rebuild method.	pass
	numbers of inputs.	1) Check that the last entry in the history of 'out' corresponds to 'matrix'. 2) Check that the re- built objects are the same	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



matrix/matrix			
	Tests that the constructor	Tests that the constructor	pass
00	method doesn't apply history to	method doesn't apply history to	
	the read MAT-file constructor.	the read MAT-file constructor.	
		1) Check that the history is the	pass
		same as the history of the saved	
		object. Because save and load	
		shouldn't add a history step. 2)	
		Check that the 'rebuild' method	
		produces the same object as	
		'out'.	
	Tests that the constructor	Tests that the constructor prop-	nass
61	properly applies history to the	erly applies history to the read	pass
	read XML-file constructor	XML-file constructor	
	read ANIE-me constructor.	1) Check that the history is the	nass
		same as the history of the saved	pass
		same as the instory of the saved	
		shouldn't add a history stop 2)	
		Check that the 'rebuild' method	
		check that the rebuild method	
		produces the same object as	
		The second secon	
62	Tests that the constructor	Tests that the constructor prop-	pass
	properly applies history in the	erly applies history in the struct	
	struct constructor.	constructor.	
		1) Check that the last entry in	pass
		the history of out corresponds	
		to the class name. 2) Check that	
		the 'rebuild' method produces	
		the same object as 'out'.	
64	Tests that the contructor	Tests that the contructor prop-	pass
	properly applies history to the	erly applies history to the	
	plist(filename) constructor.	plist(filename) constructor.	
		1) Check that the save method	pass
		doesn't change the input ob-	
		ject 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
65	Tests that the contructed object	Tests that the contructed object	pass
	can be submitted and retrieved.	can be submitted and retrieved.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to the class name. 2) Check that	
		the 'rebuild' method produces	
		the same object as 'out'.	
60	Tests that the constructor	Tests that the constructor	pass
68	properly applies history to the	properly applies history to the	
	conn+Id constructor.	conn+Id constructor.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

matrix/matrix			
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to class name. 2) Check that the	
		'rebuild' method produces the	
		same object as 'out'.	
70	Tests that the constructor	Test that the output can be	pass
10	properly applies history to the	processed back with the rebuild	
	plist(<plist-object>)</plist-object>	method.	
	constructor.	1) Check that the 'rebuild'	pass
		method produces the same ob-	
		ject as 'out'.	

Table 170: Unit tests for matrix/matrix.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

matrix/ncols			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the nools method	The noils method doesn't work	pass
02	works with a vector of	with a vector of MATRICES.	
	MATRICES as input.	Nothing to do	
			pass
03	Tests that the nools method	The noils method doesn't work	pass
	works with a matrix of	with a matrix of MATRICES.	
	MATRICES as input.	Nothing to do.	
			pass
04	Tests that the nools method	The noils method doesn't work	pass
	works with a list of MATRICES	with a list of MATRICES. Noth-	
	as input.	ing to do.	
			pass
05	Tests that the nools method	The noils method can only re-	pass
00	works with a mix of different	turn the noils values of one MA-	
	shaped MATRICES as input.	TRIX. Nothing to do	
			pass
06	Tests that the nools method	The ncols method doesn't change	pass
	properly applies history.	the MATRIX, thus will no his-	
		tory added. Nothing to do	
			pass
07	Tests that the nools method	Tests that the nools method re-	pass
	returns the number of columns	turns the number of columns of	
	of the inner objects.	the inner objects.	
		1) Check that n1 and n2 are cor-	pass
		rect.	

Table 171: Unit tests for matrix/ncols.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

matrix/nrows			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the nrows method works with a vector of MATRICES as input.	The nrows method doesn't work with a vector of MATRICES. Nothing to do	pass
			pass
03	Tests that the nrows method works with a matrix of MATRICES as input.	The nrows method doesn't work with a matrix of MATRICES. Nothing to do.	pass
			pass
04	Tests that the nrows method works with a list of MATRICES as input.	The nrows method doesn't work with a list of MATRICES. Noth- ing to do.	pass
	and the form		pass
05	Tests that the nrows method works with a mix of different shaped MATRICES as input.	The nrows method can only re- turn the nrows values of one MA- TRIX. Nothing to do	pass
			pass
06	Tests that the nrows method properly applies history.	The nrows method doesn't change the MATRIX, thus will no history added. Nothing to do	pass
			pass
07	Tests that the nrows method returns the number of columns of the inner objects.	Tests that the nrows method re- turns the number of columns of the inner objects.	pass
		1) Check that n1 and n2 are correct.	pass

Table 172: Unit tests for matrix/nrows.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

matrix/osize			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	individually	
		1) Check that getInfo call re-	nase
		turned an minfo object in all	pass
		cases. 2) Check that all plists	
		have the correct parameters.	
0.0	Tests that the osize method	The osize method doesn't work	pass
02	works with a vector of	with a vector of MATRICES.	
	MATRICES as input.	Nothing to do	
			pass
03	Tests that the osize method	The osize method doesn't work	pass
	works with a matrix of	with a matrix of MATRICES.	
	MATRICES as input.	Nothing to do.	
			pass
04	Tests that the osize method	The osize method doesn't work	pass
	works with a list of MAIRICES	with a list of MATRICES. Noth-	
	as input.	ing to do.	naga
	Tests that the esize method	The ogize method can only re-	pass
05	works with a mix of different	turn the osize values of one MA	pass
	shaped MATRICES as input	TBIX Nothing to do	
			pass
	Tests that the osize method	The osize method doesn't change	pass
06	properly applies history.	the MATRIX, thus will no his-	1
		tory added. Nothing to do	
			pass
07	Tests that the osize method	Tests that the osize method re-	pass
	returns the number of columns	turns the number of columns of	
	of the inner objects.	the inner objects.	
		1) Check that n1 and n2 are cor-	pass
		rect.	

Table 173: Unit tests for matrix/osize.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



matrix/setObjs			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.0	Tests that the [matrix/setObjs]	Test that the [matrix/setObjs]	pass
02	method works with a vector of	method works for a vector of ob-	-
	objects as input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	1
		'vec' 2) Check that each output	
		object contains the correct data.	
0.0	Tests that the [matrix/setObjs]	Test that the [matrix/setObjs]	pass
03	method works with a matrix of	method works for a matrix of ob-	1
	objects as input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	1
		'mat' 2) Check that each output	
		object contains the correct data.	
	Tests that the [matrix/setObis]	Test that the [matrix/setObis]	pass
04	method works with a list of	method works for a list of objects	1
	objects as input.	as input.	
	0 1	1) Check that the number of el-	pass
		ements in 'out' is the same as in	1
		'mat' 2) Check that each output	
		object contains the correct data.	
	Tests that the [matrix/setObis]	Tests that the [matrix/setObis]	pass
05	method works with a mix of	method works with a mix of dif-	P
	different arrays of objects as	ferent arrays of objects as input.	
	input.	1) Check that the number of el-	pass
	1	ements in 'out' is the same as in	1
		'mat' 2) Check that each output	
		object contains the correct data.	
0.0	Tests that the [matrix/setObjs]	Test that the result of applying	pass
06	method properly applies history.	the [matrix/setObjs] method can	1
		be processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[matrix/setObis]'. 2) Check	
		that the re-built object is the	
		same object as the input.	
	Tests that the setObis method	Test that the setObis method	pass
07	can modify the input MATRIX.	can modify the input MATRIX	F 3.00
		by calling with no output.	
		output	1



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

matrix/setObjs			
		1) Check that 'out' and 'obj_eq'	pass
		are now different. 2) Check that	
		'obj_eq' is not changed 3) Check	
		that the modified input is the se-	
		tObjs value of the copy 4) Check	
		that out and amodi are the same	
00	Tests that the setObjs method	Test that the setObjs method	pass
08	can set the property without a	can modify the property 'objs'	
	plist.	without a plist.	
		1) Check that 'out' has the cor-	pass
		rect objs field 2) Check that the	
		re-built object is the same object	
		as 'out'.	
00	Check that the setObjs method	Call the method with a list of	pass
09	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
11	Check that the [matrix/setObjs]	Call the method with a list of	pass
	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 174: Unit tests for matrix/setObjs.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



matrix/transpose			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the transpose method works with a vector of MATRICES as input.	Test that the transpose method works for a vector of MATRICES as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mav' 2) Check that each output contains the correct data.	pass
03	Tests that the transpose method works with a matrix of MATRICES as input.	Test that the transpose method works for a matrix of MATRI- CES as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mam' 2) Check that each output contains the correct data.	pass
04	Tests that the transpose method works with a list of MATRICES as input.	Test that the transpose method works for a list of MATRICES as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the transpose method works with a mix of different shaped MATRICES as input.	Test that the transpose method works with an input of matrices and vectors and single MATRI- CES.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the transpose method properly applies history.	Test that the result of apply- ing the transpose method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'transpose'. 2) Check that the re-built object is the same object as 'out'.	pass
07	Tests that the transpose method can modify the input MATRIX.	Test that the transpose method can modify the input MATRIX by calling with no output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

matrix/transpose			
		1) Check that 'out' and 'obj_eq'	pass
		are now different. 2) Check that	
		'obj_eq' is not changed 3) Check	
		that the modified input is the	
		transpose value of the copy 4)	
		Check that out and amodi are	
		the same	
08	Tests that the transpose method	Tests that the transpose method	pass
00	accepts plist as an input.	accepts plist as an input.	
		1) Check that the correct data 2)	pass
		Check that the re-built object is	
		the same object as 'out'.	
11	Check that the	Call the method with a list of	pass
	[matrix/transpose] method pass	output variables and with a sin-	
	back the output objects to a list	gle output variable. Additionaly	
	of output variables or to a single	check that the rebuild method	
	variable.	works on the output.	
		1) Check that the output con-	pass
		tains the same plotinfo plist	

Table 175: Unit tests for matrix/transpose.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

mfir/char			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the char method	Test that the char method works	pass
02	works with a vector of MFIR	for a vector of MFIR objects as	
	objects as input.	input.	
		1) Check that the output contain	pass
		at least each object name	
02	Tests that the char method	Test that the char method works	pass
05	works with a matrix of MFIR	for a matrix of MFIR objects as	
	objects as input.	input.	
		1) Check that the output contain	pass
		at least each object name	
04	Tests that the char method	Test that the char method works	pass
04	works with a list of MFIR	for a list of MFIR objects as in-	
	objects as input.	put.	
		1) Check that the output contain	pass
		at least each object name	
05	Tests that the char method	Test that the char method works	pass
00	works with a mix of different	with an input of matrices and	
	shaped MFIR objects as input.	vectors and single MFIR objects.	
		1) Check that the output contain	pass
		at least each object name	
06	Tests that the char method	The method char doesn't change	pass
00	properly applies history.	the data, thus it is not possible	
		to check the history. Nothing to	
		do.	
			pass

Table 176: Unit tests for mfir/char.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

mfir/copy			
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	

Table 177: Unit tests for mfir/copy.


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mfir/created			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the created method	Test that the created method	pass
	works with a vector of MFIR	works for a vector of MFIR ob-	
	objects as input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'hrv' 2) Check that each output	
		contains the correct data.	
03	Tests that the created method	Test that the created method	pass
	works with a matrix of MFIR	works for a matrix of MFIR ob-	
	objects as input.	Jects as input.	
		1) Uneck that the number of el-	pass
		'firm' 2) Check that each output	
		approximate the compatible data	
	Tests that the exected method	Tast that the greated method	naga
04	works with a list of MEID	works for a list of MEID abjects	pass
	objects as input	works for a list of MFTR objects	
	objects as input.	1) Check that the number of el	nase
		ements in 'out' is the same as in	pass
		input 2) Check that each output	
		contains the correct data	
	Tests that the created method	Test that the created method	nass
05	works with a mix of different	works with an input of matrices	pass
	shaped MFIR objects as input	and vectors and single MFIB ob-	
	shaped in it objects as input.	iects.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	P
		input. 2) Check that each output	
		contains the correct data.	
00	Tests that the created method	This method doesn't change the	pass
00	properly applies history	input object, thus no history is	
	· · · · · · ·	added to the object.	
		1) Nothing to check.	pass
07	Tests that the created method	Tests that the created method	pass
	can be used with the modify	can be used with the modify	
	command.	command.	
		1) Check the single object 2)	pass
		Check the matrix object	
08	Tests that the created method	Test that the created method	pass
00	retruns always a well defined	with an empty 'MFIR object	
	time object even for an empty	1) Check that the output is a	pass
	input object.	time object with a ell defined	
		time.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mfir/created	l
--------------	---

Table 178: Unit tests for mfir/created.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mir/creator			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the creator method	Test that the creator method	pass
	works with a vector of MFIR	works for a vector of MFIR ob-	
	objects as input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'firv' 2) Check that each output	
		contains the correct data.	
03	Tests that the creator method	Test that the creator method	pass
	works with a matrix of MFIR	works for a matrix of MFIR ob-	
	objects as input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'firm' 2) Check that each output	
		contains the correct data.	
04	Tests that the creator method	The creator method doesn't work	pass
01	works with a list of MFIR	for a list of MFIR objects as in-	
	objects as input.	put.	
		1) Nothing to test.	pass
05	Tests that the creator method	The creator method doesn't work	pass
	works with a mix of different	with different shaped input ob-	
	shaped MFIR objects as input.	jects.	
		1) Nothing to test	pass
06	Tests that the creator method	This method doesn't change the	pass
	properly applies history	input object, thus no history is	
		added to the object.	
		1) Nothing to check.	pass
07	Tests that the creator method	Tests that the creator method	pass
	can be used with the modify	can be used with the modify	
	command.	command.	
		1) Check the single object 2)	pass
		Check the matrix object	
08	Tests that the creator method	Test that the creator method	pass
	retruns all	uses the option 'all' direct or in	
	creator(s)/modifier(s) which are	a plist. The test file must have	
	in the history.	the modifier 'first', 'second' and	
		'third'	
		1) Check that out1 contains only	pass
		one creator 2) Check that out2	
		contain more creator/modifier	
09	Tests the negative case for the	Test that the creator method	pass
	option 'all'.	throws an error if the option 'all'	
		is used in connection with a ma-	
		trix/vector of MFIR objects.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mfir/creator		
	1) Nothing to test.	pass

Table 179: Unit tests for mfir/creator.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

mfir/display			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the display method	Test that the display method	pass
02	works with a vector of MFIR	works for a vector of MFIR ob-	
	objects as input.	jects as input.	
		1) Check that the output contain	pass
		at least each object name	
03	Tests that the display method	Test that the display method	pass
00	works with a matrix of MFIR	works for a matrix of MFIR ob-	
	objects as input.	jects as input.	
		1) Check that the output contain	pass
		at least each object name	
04	Tests that the display method	Test that the display method	pass
01	works with a list of MFIR	works for a list of MFIR objects	
	objects as input.	as input.	
		1) Check that the output contain	pass
		at least each object name	
05	Tests that the display method	Test that the display method	pass
	works with a mix of different	works with an input of matrices	
	shaped MFIR objects as input.	and vectors and single MFIR ob-	
		jects as.	
		1) Check that the output contain	pass
		at least each object name	
06	Tests that the display method	The method display doesn't	pass
	properly applies history.	change the data, thus it is not	
		possible to check the history.	
		Nothing to do.	
			pass

Table 180: Unit tests for mfir/display.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mfir/eq			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the eq method works with a vector of MFIR objects as input.	Test that the eq method works for a vector of MFIR objects as input. Test the positive and the negative case.	pass
		function.	pass
03	Tests that the eq method works with a matrix of MFIR objects as input.	Test that the eq method works for a matrix of MFIR objects as input. Test the positive and the negative case.	pass
		1) Check the output of the eq function.	pass
04	Tests that the eq method works with a list of MFIR objects as input.	The eq method doesn't works for a list of MFIR objects as input. Nothing to do.	pass
			pass
05	Tests that the eq method works with a mix of different shaped MFIR objects as input.	The eq method doesn't works for a list of MFIR objects as input. Nothing to do.	pass
			pass
06	Tests that the eq method properly applies history.	The eq method doesn't change the MFIR object, thus will no history added. Nothing to do	pass
			pass
07	Test the eq method with an exception list. With the LTPDA toolbox 2.0 it is only possible to test the exception list with properties where a public set method exist.	Test the eq method with the exception 'name'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because fir is created at an other time. 1) Check the output.	pass
08	Test the eq method with an	Test the eq method with the ex-	pass
	exception list. With the LTPDA toolbox 2.0 it is only possible to test the exception list with properties where a public set method exist.	 ception 'histout'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because fir is created at an other time. 1) Check the output. 	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

mfir/eq			
00	Test the eq method with an	Test the eq method with the ex-	pass
09	exception list. With the LTPDA	ception 'iunits'. Use the option	
	toolbox 2.0 it is only possible to	'internal' to suppress the history.	
	test the exception list with	It is necessary to add 'created' to	
	properties where a public set	the exception list because fir is	
	method exist.	created at an other time.	
		1) Check the output.	pass
10	Test the eq method with an	Test the eq method with the ex-	pass
10	exception list. With the LTPDA	ception 'ounits'. Use the option	
	toolbox 2.0 it is only possible to	'internal' to suppress the history.	
	test the exception list with	It is necessary to add 'created' to	
	properties where a public set	the exception list because fir is	
	method exist.	created at an other time.	
		1) Check the output.	pass
11	Test the eq method with an	Test that the eq method uses the	pass
	exception list which is in a plist.	exception list in a plist.	
		1) Check the output.	pass

Table 181: Unit tests for mfir/eq.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

mfir/get			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests the get method of the mfir	Test that the get returns returns	pass
02	class.	the value of the specified prop-	
		erty. Do this for all properties of	
		the MFIR object.	
		1) Check the correct value of the	pass
		output	
03	Tests that the get method works	Test that the get returns returns	pass
05	with a plist.	the value of the specified prop-	
		erty which is defined in a plist.	
		1) Check the correct value of the	pass
		output	
04	Tests the get method of the mfir	Test that the get throws an error	pass
04	class.	if the input are more than one	
		MFIR object.	
		1) Nothing to test	pass

Table 182: Unit tests for mfir/get.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mfir/index			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the index method works with a vector of MFIR objects as input.	Test that the index method works for a vector of MFIR ob- jects as input. The following in- dexing should work: $I = [1 \ 2 \ 3]$ or $(I/J) = [(1,1), (1,2), (1,3)]$ 1) Check that the index method	pass
03	Tests that the index method works with a matrix of MFIR objects as input.	selects the correct object. Test that the index method works for a matrix of MFIR ob- jects as input. The following in- dexing should work: $I = [1 \ 3 \ 5]$ or $(I/J) = [(1,1), (1,2), (1,3)] [2$ $4 \ 6] [(2,1), (2,2), (2,3)]$	pass
04	Tests that the index method works with a list of MFIR objects as input.	1) Check that the index method selects the correct object.The index method doesn't work for a list of MFIR objects as in- put.	pass
05	Tests that the index method properly applies history.	 Nothing to test. Test that the result of index have an additional history step. Check that the last entry in the history of 'out' corresponds to 'index' 	pass pass pass
06	Tests that the index method works for the modifier command.	Tests that the index method works for the modifier command. 1) Check that the history-plist contains the used indices. 2) Check that the index method se- lects the correct object	pass pass
07	Control the method with a plist.	 Test that the index method can be controled with a plist. 1) Check that the history-plist contains the used indices. 2) Check that the index method selects the correct object 	pass
08	Test that the index method selects more objects if I have more indices.	Test that the index method se- lects more objects if I have more indices. 1) Check that the history-plist contains the used indices. 2) Check that the index method se- lects the correct object	pass pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mfir/index

Table 183: Unit tests for mfir/index.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mfir/isprop			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the isprop method works with a vector of MFIR objects as input.	Test that the isprop method works for a vector of MFIR ob- jects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'firv' 2) Check that each output contains the correct data.	pass
03	Tests that the isprop method works with a matrix of MFIR objects as input.	Test that the isprop method works for a matrix of MFIR ob- jects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'firm' 2) Check that each output contains the correct data.	pass
04	Tests that the isprop method works with a list of MFIR objects as input.	Test that the isprop method works for a list of MFIR objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the isprop method works with a mix of different shaped MFIR objects as input.	Test that the isprop method works with an input of matrices and vectors and single MFIR ob- jects.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the isprop method properly applies history.	The method isprop doesn't change the object, thus it is not necessary to apply history.	pass
07	Tests that the isprop method works for each property.	Test that the isprop method works for the properties: 'gd', 'ntaps', 'fs', 'infile', 'a', 'histout', 'iunits', 'ounits', 'hist', 'name'	pass
08	Test the negative case and the	The contract each output contract tains the correct data. Test that the isprop method	pass
	not function command.	retrun false for a unknown prop- erty and for methods of the ob- ject.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mfir/isprop		
	1) Check that each output con-	pass
	tains the correct data.	

Table 184: Unit tests for mfir/isprop.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mfir/loadobj			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check the shape of the loaded	pass
		objects.	

Table 185: Unit tests for mfir/loadobj.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mfir/mfir			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the mfir method works with a vector of MFIR objects as input.	Test that the mfir method works with a vector of MFIR objects as input.	pass
		1) Check that the shape of the output MFIRs is the same as the input shape. 2) Check that each output MFIR is a copy of the input MFIR. 3) Check that the copy have an additional history step.	pass
03	Tests that the mfir method works with a matrix of MFIR objects as input.	Test that the mfir method works with a matrix of MFIR objects as input.	pass
		1) Check that the shape of the output MFIRs is the same as the input shape. 2) Check that each output MFIR is a copy of the input MFIR. 3) Check that the copy have an additional history step.	pass
04	Tests that the mfir method works with a list of MFIR objects as input.	Test that the mfir method works with a list of MFIR objects as in- put.	pass
		1) Check that the number of ele- ments in 'out' is the same of the number in the input. 2) Check that each output MFIR is a copy of the input MFIR. 3) Check that the copy have an additional his- tory step.	pass
05	Tests that the mfir method works with a mix of different shaped MFIRs as input.	Test that the mfir method works with a mix of different shaped MFIRs as input.	pass
		1) Check that the number of ele- ments in 'out' is the same of the number in the input. 2) Check that each output MFIR is a copy of the input MFIR. 3) Check that the copy have an additional his- tory step.	pass
06	Tests that the mfir method properly applies history.	Test that the result of apply- ing the mfir method can be pro- cessed back.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mfir/mfir			
	Tests that the mfir method	 Check that the last entry in the history of 'out' corre- sponds to 'mfir'. 2) Check that the method rebuild produces the same object as 'out'. Test that the output can be pro- 	pass
07	properly applies history to the copy constructor.	cessed back with the 'rebuild' method. Test the constructor with a different number of in- puts.	-
		1) Check that the last entry in the history of 'out' corresponds to 'mfir'. 2) Check that the orig- inal objects are not changed by the setter function 3) Check that the method rebuild produces the same object as 'out'.	pass
08	Tests that the mfir method properly applies history to the read MAT-file constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'mfir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
09	Tests that the mfir method properly applies history to the read XML-file constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check the shape 2) Check that the last entry in the history of 'out' corresponds to 'mfir'. 3) Check that the 'rebuild' method produces the same object as 'out'.	pass
10	Tests that the mfir method properly applies history to the AO constructor.	Test that the output can be processed back with the 're- build' method. Use the default values (method = 'frequency- sampling').	pass
		1) Check that the last entry in the history of 'out' corresponds to 'mfir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
11	Tests that the mfir method properly applies history to the struct constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mfir/mfir			
		1) Check that the last entry in the history of 'out' corresponds to 'mfir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
12	Tests that the mfir method properly applies history to the pzmodel constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'mfir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
13	Tests that the mfir method properly applies history to the plict (filmane) constructor	Test that the output can be pro- cessed back to an m-file.	pass
	plist(filename) constructor.	1) Check that the save method doesn't change the input ob- ject 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
14	Tests that the MFIR method properly applies history to the plist(conn) constructor.	Test that the output can be processed back with the rebuild method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'mfir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
15	Tests that the MFIR method properly applies history to the plist(type) constructor.	Test that the output can be processed back with the rebuild method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'mfir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
16	Tests that the MFIR method properly applies history to the plist(pzmodel) constructor.	Test that the output can be processed back with the rebuild method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'mfir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
17	Tests that the MFIR method properly applies history to the plist(<plist-object>) constructor.</plist-object>	Test that the output can be processed back with the rebuild method.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

mfir/mfir			
		1) Check that the last entry in the history of 'out' corresponds to 'mfir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
18	Tests that the MFIR method properly applies history to the a constructor.	Test that the output can be processed back with the rebuild method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'mfir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
19	Tests that the mfir method properly applies history to the plist(AO) constructor.	Test that the output can be processed back with the 're- build' method. Use the method 'frequency-sampling'.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'mfir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
20	Tests that the mfir method properly applies history to the plist(AO) constructor.	Test that the output can be pro- cessed back with the 'rebuild' method. Use the method 'least- squares'.	pass
		method an error.	pass
21	Tests that the mfir method properly applies history to the plist(AO) constructor.	Test that the output can be pro- cessed back with the 'rebuild' method. Use the method 'Parks- McClellan'.	pass
		1) At the moment throws this method an error.	pass
22	Tests that the mfir method properly applies history to the AO + plist constructor.	Test that the output can be processed back with the 're- build' method. Use the method 'frequency-sampling'.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'mfir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
23	Tests that the MFIR method properly applies history to the conn+Id constructor.	Test that the output can be processed back with the rebuild method.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

mfir/mfir			
		1) Check that the last entry in the history of 'out' corre- sponds to 'mfir'. 2) Check that re-running the 'test.m' file pro- duces the same object as 'out'.	pass
24	Tests that the MFIR method properly applies history to the pole/zero model + plist object	Test that the output can be processed back with the rebuild method.	pass
	constructor.	1) Check that the last entry in the history of 'out' corresponds to 'mfir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
25	Tests that the MFIR method properly applies history to the a + fs object constructor.	Test that the output can be processed back with the rebuild method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'mfir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass

Table 186: Unit tests for mfir/mfir.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

mfir/ne			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInto call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the ne method works	lest that the ne method works	pass
	with a vector of MFIR objects	for a vector of MFIR objects as	
	as input.	input. Test the positive and the	
		negative case.	
		1) Check the output of the ne	pass
		function.	
03	Tests that the ne method works	lest that the ne method works	pass
	with a matrix of MFIR objects	for a matrix of MFIR objects as	
	as input.	input. Test the positive and the	
		negative case.	
		1) Check the output of the ne	pass
		function.	
04	Tests that the ne method works	The ne method doesn't works for	pass
	with a list of MFIR objects as	a list of MFIR objects as input.	
	input.	Nothing to do.	
			pass
05	lests that the ne method works	I he he method doesn't works for	pass
	MEID objects og input	Nothing to do	
	MFIR objects as input.	Nothing to do.	pagg
	Tests that the ne method	The ne method descrit shange	pass
06	properly applies history	the MFIR object thus will no	pass
	property applies instory.	history added Nothing to do	
		instory added. Nothing to do	nass
	Test the ne method with an	Test the ne method with the ex-	pass
07	exception list. The function	contion 'name' Use the option	pass
	$m_{\rm m}$	'internal' to suppress the history	
	so it is not necessary to check all	It is necessary to add 'created' to	
	possibilities of the exception list	the exception list because fir is	
		created at an other time	
		1) Check that each output con-	pass
		tains the correct data	Pass
	Test the ne method with an	Test that the ne method uses the	nass
08	exception list which is in a plist	exception list in a plist	Pass
		1) Check that each output con-	pass
		tains the correct data.	Pass
1			1

Table 187: Unit tests for mfir/ne.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

mfir/rebuild			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInto call re-	pass
		turned an minto object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the rebuild method	lest that the rebuild method	pass
	works with a vector of MFIR	works for a vector of MFIR ob-	
	objects as input.	jects as input.	
		1) Check the rebuilt output.	pass
03	Tests that the rebuild method	Test that the rebuild method	pass
	works with a matrix of MFIR	works for a matrix of MFIR ob-	
	objects as input.	jects as input.	
		1) Check the rebuilt output.	pass
04	Tests that the rebuild method	Test that the rebuild method	pass
	works with a list of MFIR	works for a list of MFIR objects	
	objects as input.	as input.	
		1) Check the rebuilt output.	pass
05	Tests that the rebuild method	Test that the rebuild method	pass
	works with a mix of different	works with an input of matrices	
	shaped MFIR objects as input.	and vectors and single MFIR ob-	
		jects.	
		1) Check the rebuilt output.	pass
06	Tests that the rebuild method	The method rebuild doesn't	pass
	properly applies history.	change the data, thus it is not	
		possible to check the history.	
		Nothing to do.	
			pass
07	Check that the rebuild method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 188: Unit tests for mfir/rebuild.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mfir/redesign			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the redesign method works with a vector of MFIR objects as input.	Test that the redesign method works for a vector of MFIR ob- jects as input. To keep this UTP simple use for the vector only one filter object. The different filters will be tested in an other UTP.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'firvec' 2) Check that each out- put MFIR contains the correct data.	pass
03	Tests that the redesign method works with a matrix of MFIR objects as input.	Test that the redesign method works for a matrix of MFIR ob- jects as input. To keep this UTP simple use for the matrix only one filter object. The different fil- ters will be tested in an other UTP.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'firmat' 2) Check that each out- put MFIR contains the correct data.	pass
04	Tests that the redesign method works with a list of MFIR objects as input.	Test that the redesign method works for a list of MFIR objects as input. To keep this UTP sim- ple use for the list only one filter object. The different filters will be tested in an other UTP.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output MFIR contains the correct data.	pass
05	Tests that the redesign method works with a mix of different shaped MFIR objects as input.	Test that the redesign method works with an input of matrices and vectors and single MFIR ob- jects. To keep this UTP simple use for the vector only one filter object. The different filters will be tested in an other UTP.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

mfir/redesign			
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		MFIR contains the correct data.	
06	Tests that the redesign method	Test that the result of applying	pass
00	properly applies history.	the redesign method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'redesign'. 2) Check that re-	
		built object is the same object as	
		the input.	
07	Tests that the redesign method	Tests that the redesign method	pass
07	redesigns a standard filter type.	redesigns a standard filter type.	
		1) Check the output 2) Check the	pass
		rebuilt object	
09	Tests that the redesign method	Tests that the redesign method	pass
08	redesigns a pzmodel filter type.	redesigns a pzmodel filter type.	
		1) Check the output 2) Check the	pass
		rebuilt object	

Table 189: Unit tests for mfir/redesign.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

m mfir/resp			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the resp method works with a vector of MFIR objects as input.	Test that the resp method works for a vector of MFIR objects as input. Test the method with an output and with no output (a di- agram must appear)	pass
		1) Test the right number of lines in the diagram. 2) Check that the number of elements in 'out' is the same as in 'firv' 3) Check that each output MFIR contains the correct data.	pass
03	Tests that the resp method works with a matrix of MFIR objects as input.	Test that the resp method works for a matrix of MFIR objects as input. Test the method with an output and with no output (a di- agram must appear)	pass
		1) Test the right number of lines in the diagram. 2) Check that the number of elements in 'out' is the same as in 'firmat' 3) Check that each output MFIR contains the correct data.	pass
04	Tests that the resp method works with a list of MFIR objects as input.	Test that the resp method works for a list of MFIR objects as in- put. Test the method with an output and with no output (a di- agram must appear)	pass
		1) Test the right number of lines in the diagram. 2) Check that the number of elements in 'out' is the same as in 'firmat' 3) Check that each output MFIR contains the correct data.	pass
05	Tests that the resp method works with a mix of different shaped MFIR objects as input.	Test that the resp method works with an input of matrices and vectors and single MFIR objects. Test the method with an output and with no output (a diagram must appear)	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mfir/resp			
	Tests that the resp method	1) Test the right number of lines in the diagram. 2) Check that the number of elements in 'out' is the same as in 'firmat' 3) Check that each output MFIR contains the correct data.	pass
06	properly applies history.	 1) Check that the last entry in the history of 'out' corresponds to 'resp'. 2) Check that re-built object is the same object as the 	pass
		input.	
07	Tests that modify command plots the response into a diagram.	Tests that modify command plots the response into a dia- gram.	pass
		Tost that the output AO of the	pass
08	Test the shape of the output.	resp method keeps the shape of the used input f vector.	pass
		1) Check that the shape of the data doesn't change.	pass
09	Check that the resp method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output. 1) Check that the output con-	pass
		tains the right number of objects 2) Check that the 'rebuild' method produces the same object as 'out'.	
10	Check that the resp method uses the x-data of an input AO for f-vector	Call the method with different method to pass an AO in.	pass
		that the 'rebuild' method pro- duces the same object as 'out'.	pass
11	Check that the resp method uses the specified f-vector to compute the response.	Call the method with different method to pass an f-vector in. 1) Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'.	pass pass
12	Check that the resp method uses the specified f1, f2, and nf to compute the response.	Call the method with different method to pass f1, f2, and nf in. 1) Check the output 2) Check	pass pass
		that the 'rebuild' method pro- duces the same object as 'out'.	T
13	Check that the resp method uses the specified f1, f2, and nf to compute the response.	Call the method with different method to pass f1, f2, and nf in.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

mfir/resp			
		1) Check the output 2) Check	pass
		that the 'rebuild' method pro-	
		duces the same object as 'out'.	
14	Check that the resp method the	Check that the resp method the	pass
14	response of a serial filter bank.	response of a serial filter bank.	
		1) Check the output 2) Check	pass
		that the 'rebuild' method pro-	
		duces the same object as 'out'.	
15	Check that the resp method the	Check that the resp method the	pass
10	response of a parallel filter bank.	response of a parallel filter bank.	
		1) Check the output 2) Check	pass
		that the 'rebuild' method pro-	
		duces the same object as 'out'.	

Table 190: Unit tests for mfir/resp.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mfir/save			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the save method works with a vector of MFIR objects as input.	Test that the save method works for a vector of MFIR objects as input. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of ele- ments in 'out1' and 'out2' are the same as in 'firv' 2) Check that the loaded objects are the same as the saved objects. 3) The out- puts 'out1' and 'out2' must be the same.	pass
03	Tests that the save method works with a matrix of MFIR objects as input.	Test that the save method works for a matrix of MFIR objects as input. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out1' and 'out2' are the same as in 'firm' 2) Check that the loaded objects are the same as the saved objects. 3) The outputs 'out1' and 'out2' must be the same.	pass
04	Tests that the save method works with a list of MFIR objects as input.	Test that the save method works for a list of MFIR objects as in- put. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out1' and 'out2' are the same as in the list 2) Check that the loaded objects are the same as the saved objects. 3) The outputs 'out1' and 'out2' must be the same.	pass
05	Tests that the save method works with a mix of different shaped MFIR objects as input.	Test that the save method works with an input of matrices and vectors and single MFIR ob- jects. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output MFIR object contains the correct data.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

mfir/save			
06	Tests that the save method properly applies history.	Test that the result of apply- ing the save method can be pro- cessed back to an m-file. Do this for both extensions 'mat' and 'xml'	pass
		 Check that the history applies to the output object. Check that save doesn't add a history step to the input object. 2) Check that the read object doesn't contain the save + load history steps. Check that the method re- build produces the same object as 'out'. 	pass
07	Tests that the save method works with the modify	Use the save method with the modifier command.	pass
	command.	1) Check that the save method applies the history. 2) Check the output against the input.	pass
08	Control the method with a plist.	Test that the save method uses the filename which is stored in a plist.	pass
		1) Check the output	pass
09	Test the save method with standard MFIR objects.	Save all standard MFIR objects with both extensions.	pass
		1) Check the output	pass
10	Test the save method with MFIR object which is created	Save MFIR object which is cre- ated from a pzmodel.	pass
	from a pole/zero model	1) Check the output	pass
11	Test the save method with MFIR object which is created	Save MFIR object which is cre- ated from an analysis model.	pass
	from an analysis model.	1) Check the output	pass

Table 191: Unit tests for mfir/save.

~



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

mfir/setHistout			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the setHistout method works with a vector of MFIR objects as input.	The setHistout should set the output history (histout) of each input.	pass
		1) Check the histout has the correct values	pass
03	Tests that the setHistout method works with a matrix of MFIR objects as input.	The setHistout should set the output history (histout) of each input.	pass
		1) Check the histout has the correct values	pass
04	Tests that the setHistout method works with a list of MFIR objects as input.	The setHistout should set the output history (histout) of each input.	pass
		1) Check the histout has the correct values	pass
05	Tests that the setHistout method works with a mix of different shaped MFIR objects	The setHistout method is not de- signed for this call, for that rea- son must this call fail.	pass
	as input.	1) Check the histout has the correct values	pass
06	Tests that the setHistout method properly applies history and that the option 'internal'	Test that the result of apply- ing the setHistout method can be processed back to an m-file.	pass
	suppresses the history.	1) Check that the last entry in the history of 'out1' corre- sponds to 'setHistout'. 2) Check that the last entry in the his- tory of 'out2' NOT corresponds to 'setHistout'. 3) Check that the 'rebuild' method produces the same object as 'out'.	pass
07	Tests that the setHistout method can modify the input MFIR object.	Test that the setHistout method can modify the input MFIR ob- ject by calling with no output.	pass
		now different. 2) Check that 'ain' has the correct histout field	pass
08	Tests that the setHistout method can set the property with a plist.	Test that the setHistout method can modify the property 'histout' with a value in a plist.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mfir/setHistout		
	1) Check that 'ain' has the cor-	pass
	rect histout field 2) Check that	
	the 'rebuild' method produces	
	the same object as 'out'.	

Table 192: Unit tests for mfir/setHistout.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mfir/setIunits			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all	pass
		have the correct parameters.	
02	Tests that the setIunits method works with a vector of MFIR	Test that the setIunits method works for a vector of MFIR ob-	pass
	objects as input.	 jects as input. 1) Check that the number of elements in 'out' is the same as in 	pass
		'firv' 2) Check that each output contains the correct data.	
03	Tests that the setIunits method works with a matrix of MFIR	Test that the setIunits method works for a matrix of MFIR ob-	pass
	objects as input.	 Check that the number of elements in 'out' is the same as in 'firm' 2) Check that each output contains the correct data 	pass
04	Tests that the setIunits method works with a list of MFIR objects as input.	Test that the setIunits method works for a list of MFIR objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the setIunits method works with a mix of different shaped MFIR objects as input.	Test that the setIunits method works with an input of matrices and vectors and single MFIR ob- jects.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the setIunits method properly applies history and that the option 'internal'	Test that the result of applying the setIunits method can be pro- cessed back to an m-file.	pass
	suppresses the history.	1) Check that the last entry in the history of 'out1' corre- sponds to 'setIunits'. 2) Check that the last entry in the history of 'out2' NOT corresponds to 'setIunits'. 3) Check that the 're- build' method produces the same object as 'out'.	pass
07	Tests that the setlunits method can modify the input MFIR object.	Test that the setlunits method can modify the input MFIR ob- ject by calling with no output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

mfir/setIunits			
		1) Check that 'firhp' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct iunits field	
08	Tests that the setIunits method	Test that the setIunits method	pass
00	can set the property with a	can modify the property 'iunits'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect iunits field 2) Check that the	
		'rebuild' method produces the	
		same object as 'out'.	
00	Check that the setIunits method	Call the method with a list of	pass
09	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 193: Unit tests for mfir/setIunits.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mfir/setName			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the setName method	Test that the setName method	pass
02	works with a vector of MFIR	works for a vector of MFIR ob-	
	objects as input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'firv' 2) Check that each output	
		contains the correct data.	
0.0	Tests that the setName method	Test that the setName method	pass
03	works with a matrix of MFIR	works for a matrix of MFIR ob-	
	objects as input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	-
		'firm' 2) Check that each output	
		contains the correct data.	
	Tests that the setName method	Test that the setName method	pass
04	works with a list of MFIR	works for a list of MFIR objects	P
	objects as input.	as input.	
	r g · · · g · · · · · · · · · · · · · ·	1) Check that the number of el-	pass
		ements in 'out' is the same as in	T
		input. 2) Check that each output	
		contains the correct data.	
	Tests that the setName method	Test that the setName method	pass
05	works with a mix of different	works with an input of matrices	T
	shaped MFIR objects as input.	and vectors and single MFIR ob-	
		jects.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	T
		input. 2) Check that each output	
		contains the correct data.	
	Tests that the setName method	Test that the result of applying	pass
06	properly applies history and	the setName method can be pro-	T
	that the option 'internal'	cessed back to an m-file.	
	suppresses the history.	1) Check that the last entry	pass
		in the history of 'out1' corre-	P
		sponds to 'setName'. 2) Check	
		that the last entry in the history	
		of 'out2' NOT corresponds to	
		'setName', 3) Check that the 're-	
		build' method produces the same	
		object as 'out'	
	Tests that the setName method	Test that the setName method	nass
07	can modify the input MFIR	can modify the input MFIR ob	Pass
	object	iect by calling with no output	
	00,000.	Jees by canning with no output.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

mfir/setName			
		1) Check that 'firhp' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct name field	
08	Tests that the setName method	Test that the setName method	pass
00	can set the property with a	can modify the property 'name'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect name field 2) Check that the	
		'rebuild' method produces the	
		same object as 'out'.	
00	Check that the setName method	Call the method with a list of	pass
09	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 194: Unit tests for mfir/setName.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



mfir/setOunits			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the setOunits method works with a vector of MFIR objects as input.	Test that the setOunits method works for a vector of MFIR ob- jects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'firv' 2) Check that each output contains the correct data.	pass
03	Tests that the setOunits method works with a matrix of MFIR objects as input.	Test that the setOunits method works for a matrix of MFIR ob- jects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'firm' 2) Check that each output contains the correct data.	pass
04	Tests that the setOunits method works with a list of MFIR objects as input.	Test that the setOunits method works for a list of MFIR objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the setOunits method works with a mix of different shaped MFIR objects as input.	Test that the setOunits method works with an input of matrices and vectors and single MFIR ob- jects.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the setOunits method properly applies history and that the option 'internal'	Test that the result of apply- ing the setOunits method can be processed back to an m-file.	pass
	suppresses the history.	1) Check that the last entry in the history of 'out1' corre- sponds to 'setOunits'. 2) Check that the last entry in the history of 'out2' NOT corresponds to 'se- tOunits'. 3) Check that the 're- build' method produces the same object as 'out'.	pass
07	Tests that the setOunits methodcan modify the input MFIRobject.	Test that the setOunits method can modify the input MFIR ob- ject by calling with no output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

mfir/setOunits			
		1) Check that 'firhp' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct ounits field	
08	Tests that the setOunits method	Test that the setOunits method	pass
00	can set the property with a	can modify the property 'ounits'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect ounits field 2) Check that	
		the 'rebuild' method produces	
		the same object as 'out'.	
00	Check that the setOunits	Call the method with a list of	pass
09	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 195: Unit tests for mfir/setOunits.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

	Tests that the getInfo call works	Test that the getInfo call works	m a a a
01	for this method.	for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the string method works with a vector of MFIR objects as input.	Test that the string method works for a vector of MFIR ob- jects as input.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
03	Tests that the string method works with a matrix of MFIR objects as input.	Test that the string method works for a matrix of MFIR ob- jects as input.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
04	Tests that the string method works with a list of MFIR objects as input.	Test that the string method works for a list of MFIR objects as input.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
05	Tests that the string method works with a mix of different shaped MFIR objects as input.	Test that the string method works with an input of matrices and vectors and single MFIR ob- jects.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
06	Tests that the string method properly applies history.	The method string doesn't change the data, thus it is not possible to check the history. Nothing to do.	pass
07	Tests that the string method doesn't work if the MFIR object	The method string throws an er-	pass
	have more than one history step.	more than one history step.	pass

Table 196: Unit tests for mfir/string.


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

mfir/type			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.0	Tests that the type method	Test that the type method works	pass
02	works with a vector of MFIR	for a vector of MFIR objects as	
	objects as input.	input.	
		1) Check the rebuilt output.	pass
02	Tests that the type method	Test that the type method works	pass
05	works with a matrix of MFIR	for a matrix of MFIR objects as	
	objects as input.	input.	
		1) Check the rebuilt output.	pass
04	Tests that the type method	Test that the type method works	pass
04	works with a list of MFIR	for a list of MFIR objects as in-	
	objects as input.	put.	
		1) Check the rebuilt output.	pass
05	Tests that the type method	Test that the type method works	pass
00	works with a mix of different	with an input of matrices and	
	shaped MFIR objects as input.	vectors and single MFIR objects.	
		1) Check the rebuilt output.	pass
06	Tests that the type method	The method type doesn't change	pass
00	properly applies history.	the data, thus it is not possible	
		to check the history. Nothing to	
		do.	
			pass

Table 197: Unit tests for mfir/type.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

miir/char			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the char method	Test that the char method works	pass
02	works with a vector of MIIR	for a vector of MIIR objects as	
	objects as input.	input.	
		1) Check that the output contain	pass
		at least each object name	
02	Tests that the char method	Test that the char method works	pass
03	works with a matrix of MIIR	for a matrix of MIIR objects as	
	objects as input.	input.	
		1) Check that the output contain	pass
		at least each object name	
04	Tests that the char method	Test that the char method works	pass
04	works with a list of MIIR	for a list of MIIR objects as in-	
	objects as input.	put.	
		1) Check that the output contain	pass
		at least each object name	
05	Tests that the char method	Test that the char method works	pass
05	works with a mix of different	with an input of matrices and	
	shaped MIIR objects as input.	vectors and single MIIR objects.	
		1) Check that the output contain	pass
		at least each object name	
06	Tests that the char method	The method char doesn't change	pass
	properly applies history.	the data, thus it is not possible	
		to check the history. Nothing to	
		do.	
			pass

Table 198: Unit tests for miir/char.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

miir/copy			
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	

Table 199: Unit tests for miir/copy.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir/created			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the created method works with a vector of MIIR objects as input.	Test that the created method works for a vector of MIIR ob- jects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'iirv' 2) Check that each output contains the correct data.	pass
03	Tests that the created method works with a matrix of MIIR objects as input.	Test that the created method works for a matrix of MIIR ob- jects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'iirm' 2) Check that each output contains the correct data.	pass
04	Tests that the created method works with a list of MIIR objects as input.	Test that the created method works for a list of MIIR objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the created method works with a mix of different shaped MIIR objects as input.	Test that the created method works with an input of matrices and vectors and single MIIR ob- jects.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the created method properly applies history	This method doesn't change the input object, thus no history is added to the object. 1) Nothing to check.	pass
07	Tests that the created method can be used with the modify command.	Tests that the created method can be used with the modify command. 1) Check the single object 2)	pass
08	Tests that the created method retruns always a well defined	Check the matrix object Test that the created method with an empty 'MIIR object	pass
	time object even for an empty input object.	1) Check that the output is a time object with a ell defined time.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir/created	
--------------	--

Table 200: Unit tests for miir/created.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir/creator			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the creator method	Test that the creator method	pass
	works with a vector of MIIR	works for a vector of MIIR ob-	
	objects as input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'mrv' 2) Check that each output	
		contains the correct data.	
03	Tests that the creator method	Test that the creator method	pass
	works with a matrix of MIIR	works for a matrix of MIIR ob-	
	objects as input.	Jects as input.	
		1) Check that the number of el-	pass
		ements in out is the same as in	
		antaing the correct date	
	Togta that the exector method	The exact on most had do can't month	N 000
04	rests that the creator method	for a list of MID abjects og in	pass
	works with a list of MIIR	for a list of MIIR objects as in-	
	objects as input.	1) Nothing to tost	nass
	Tests that the creator method	The creator method doesn't work	pass
05	works with a mix of different	with different shaped input ob-	pass
	shaped MIIB objects as input	iects	
	shaped mill objects as input.	1) Nothing to test	pass
	Tests that the creator method	This method doesn't change the	pass
06	properly applies history	input object, thus no history is	Pass
		added to the object.	
		1) Nothing to check.	pass
~~	Tests that the creator method	Tests that the creator method	pass
07	can be used with the modify	can be used with the modify	
	command.	command.	
		1) Check the single object 2)	pass
		Check the matrix object	
00	Tests that the creator method	Test that the creator method	pass
08	retruns all	uses the option 'all' direct or in	
	creator(s)/modifier(s) which are	a plist. The test file must have	
	in the history.	the modifier 'first', 'second' and	
		'third'	
		1) Check that out1 contains only	pass
		one creator 2) Check that out2	
		contain more creator/modifier	
09	Tests the negative case for the	Test that the creator method	pass
	option 'all'.	throws an error if the option 'all'	
		is used in connection with a ma-	
		trix/vector of MIIR objects.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir/creator		
	1) Nothing to test.	pass

Table 201: Unit tests for miir/creator.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

miir/display			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the display method	Test that the display method	pass
02	works with a vector of MIIR	works for a vector of MIIR ob-	
	objects as input.	jects as input.	
		1) Check that the output contain	pass
		at least each object name	
03	Tests that the display method	Test that the display method	pass
	works with a matrix of MIIR	works for a matrix of MIIR ob-	
	objects as input.	jects as input.	
		1) Check that the output contain	pass
		at least each object name	
04	Tests that the display method	Test that the display method	pass
	works with a list of MIIR	works for a list of MIIR objects	
	objects as input.	as input.	
		1) Check that the output contain	pass
		at least each object name	
05	Tests that the display method	Test that the display method	pass
	works with a mix of different	works with an input of matrices	
	shaped MIIR objects as input.	and vectors and single MIIR ob-	
		jects as.	
		1) Check that the output contain	pass
		at least each object name	
06	Tests that the display method	The method display doesn't	pass
	properly applies history.	change the data, thus it is not	
		possible to check the history.	
		Nothing to do.	
			pass

Table 202: Unit tests for miir/display.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir/eq			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the eq method works with a vector of MIIR objects as input.	Test that the eq method works for a vector of MIIR objects as input. Test the positive and the negative case.	pass
		function.	pass
03	Tests that the eq method works with a matrix of MIIR objects as input.	Test that the eq method works for a matrix of MIIR objects as input. Test the positive and the negative case.	pass
		1) Check the output of the eq function.	pass
04	Tests that the eq method works with a list of MIIR objects as input.	The eq method doesn't works for a list of MIIR objects as input. Nothing to do.	pass
			pass
05	Tests that the eq method works with a mix of different shaped MIIR objects as input.	The eq method doesn't works for a list of MIIR objects as input. Nothing to do.	pass
			pass
06	Tests that the eq method properly applies history.	The eq method doesn't change the MIIR object, thus will no his- tory added. Nothing to do	pass
			pass
07	Test the eq method with an exception list. With the LTPDA toolbox 2.0 it is only possible to test the exception list with properties where a public set method exist.	Test the eq method with the exception 'name'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because iir is created at an other time. 1) Check the output.	pass
0.0	Test the eq method with an	Test the eq method with the ex-	pass
08	exception list. With the LTPDA toolbox 2.0 it is only possible to test the exception list with properties where a public set method exist.	ception 'histin'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because iir is created at an other time. 1) Check the output	nass
1		I) Uneck the output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

miir/eq			
09	Test the eq method with an exception list. With the LTPDA toolbox 2.0 it is only possible to test the exception list with properties where a public set method exist.	Test the eq method with the exception 'histout'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because iir is created at an other time.	pass
10	Test the eq method with an exception list. With the LTPDA toolbox 2.0 it is only possible to test the exception list with properties where a public set method exist.	Test the eq method with the exception 'iunits'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because iir is created at an other time.	pass
11	Test the eq method with an exception list. With the LTPDA toolbox 2.0 it is only possible to test the exception list with properties where a public set method exist.	 Check the output. Test the eq method with the exception 'ounits'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because iir is created at an other time. Check the output. 	pass
12	Test the eq method with an exception list which is in a plist.	Test that the eq method uses the exception list in a plist.1) Check the output.	pass pass

Table 203: Unit tests for miir/eq.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

miir/get			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests the get method of the miir	Test that the get returns returns	pass
02	class.	the value of the specified prop-	
		erty. Do this for all properties of	
		the MIIR object.	
		1) Check the correct value of the	pass
		output	
02	Tests that the get method works	Test that the get returns returns	pass
05	with a plist.	the value of the specified prop-	
		erty which is defined in a plist.	
		1) Check the correct value of the	pass
		output	
04	Tests the get method of the miir	Test that the get throws an error	pass
04	class.	if the input are more than one	
		MIIR object.	
		1) Nothing to test	pass

Table 204: Unit tests for miir/get.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir/index			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the index method works with a vector of MIIR objects as input.	Test that the index method works for a vector of MIIR ob- jects as input. The following in- dexing should work: I = $[1\ 2\ 3]$ or $(I/J) = [(1,1), (1,2), (1,3)]$ 1) Check that the index method	pass
03	Tests that the index method works with a matrix of MIIR objects as input.	selects the correct object. Test that the index method works for a matrix of MIIR ob- jects as input. The following in- dexing should work: $I = [1 \ 3 \ 5]$ or $(I/J) = [(1,1), (1,2), (1,3)] [2$ 4 6] $[(2,1), (2,2), (2,3)]$ 1) Check that the index method	pass
04	Tests that the index method works with a list of MIIR objects as input.	The index method doesn't work for a list of MIIR objects as in- put.	pass
05	Tests that the index method properly applies history.	 1) Nothing to test. Test that the result of index have an additional history step. 1) Check that the last entry in the history of 'out' corresponds to 'index'. 	pass pass pass
06	Tests that the index method works for the modifier command.	Tests that the index method works for the modifier command. 1) Check that the history-plist contains the used indices. 2) Check that the index method se- lects the correct object	pass pass
07	Control the method with a plist.	 Test that the index method can be controled with a plist. 1) Check that the history-plist contains the used indices. 2) Check that the index method se- lects the correct object 	pass
08	Test that the index method selects more objects if I have more indices.	Test that the index method selects more objects if I have more indices.1) Check that the history-plist contains the used indices. 2) Check that the index method selects the correct object	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir/index

Table 205: Unit tests for miir/index.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir/isprop			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the isprop method works with a vector of MIIR objects as input.	Test that the isprop method works for a vector of MIIR ob- jects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'iirv' 2) Check that each output contains the correct data.	pass
03	Tests that the isprop method works with a matrix of MIIR objects as input.	Test that the isprop method works for a matrix of MIIR ob- jects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'iirm' 2) Check that each output contains the correct data.	pass
04	Tests that the isprop method works with a list of MIIR objects as input.	Test that the isprop method works for a list of MIIR objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the isprop method works with a mix of different shaped MIIR objects as input.	Test that the isprop method works with an input of matrices and vectors and single MIIR ob- jects.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the isprop method properly applies history.	The method isprop doesn't change the object, thus it is not necessary to apply history.	pass
07	Tests that the isprop method works for each property.	Test that the isprop method works for the properties: 'b', 'histin', 'ntaps', 'fs', 'infile', 'a', 'histout', 'iunits', 'ounits', 'hist', 'name'	pass
		tains the correct data.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir/isprop			
08	Test the negative case and the	Test that the isprop method	pass
00	not function command.	retrun false for a unknown prop-	
		erty and for methods of the ob-	
		ject.	
		1) Check that each output con-	pass
		tains the correct data.	

Table 206: Unit tests for miir/isprop.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir/loadobj			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check the shape of the loaded	pass
		objects.	

Table 207: Unit tests for miir/loadobj.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir/miir			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the miir method works with a vector of MIIR objects as input.	Test that the miir method works with a vector of MIIR objects as input.	pass
		1) Check that the shape of the output MIIRs is the same as the input shape. 2) Check that each output MIIR is a copy of the input MIIR. 3) Check that the copy have an additional history step.	pass
03	Tests that the miir method works with a matrix of MIIR objects as input.	Test that the miir method works with a matrix of MIIR objects as input.	pass
		1) Check that the shape of the output MIIRs is the same as the input shape. 2) Check that each output MIIR is a copy of the input MIIR. 3) Check that the copy have an additional history step.	pass
04	Tests that the miir method works with a list of MIIR objects as input.	Test that the miir method works with a list of MIIR objects as in- put.	pass
		1) Check that the number of ele- ments in 'out' is the same of the number in the input. 2) Check that each output MIIR is a copy of the input MIIR. 3) Check that the copy have an additional his- tory step.	pass
05	Tests that the miir method works with a mix of different shaped MIIRs as input.	Test that the miir method works with a mix of different shaped MIIRs as input.	pass
		1) Check that the number of ele- ments in 'out' is the same of the number in the input. 2) Check that each output MIIR is a copy of the input MIIR. 3) Check that the copy have an additional his- tory step.	pass
06	Tests that the miir method properly applies history.	Test that the result of apply- ing the miir method can be pro- cessed back.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir/miir			
		1) Check that the last entry in the history of 'out' corre- sponds to 'miir'. 2) Check that the method rebuild produces the same object as 'out'.	pass
07	properly applies history to the copy constructor.	Test that the output can be pro- cessed back with the 'rebuild' method. Test the constructor with a different number of in- puts.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'miir'. 2) Check that the orig- inal objects are not changed by the setter function 3) Check that the method rebuild produces the same object as 'out'.	pass
08	Tests that the miir method properly applies history to the read MAT-file constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the history is the same as the history of the saved object. Because save and load shouldn't add a history step. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
09	Tests that the miir method properly applies history to the read XML-file constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the history is the same as the history of the saved object. Because save and load shouldn't add a history step. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
10	Tests that the miir method properly applies history to the read FIL-file constructor.	Read the FIL file which is cre- ated from LISO. Test that the output can be processed back with the 'rebuild' method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'miir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
11	Tests that the miir method properly applies history to the struct constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir/miir			
		1) Check that the last entry in the history of 'out' corresponds to 'miir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
12	Tests that the miir method properly applies history to the parfrac constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'miir'. 2) Check the correct number of outputs. 3) Check that the 'rebuild' method produces the same object as 'out'.	pass
13	Tests that the miir method properly applies history to the pzmodel constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'miir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
14	Tests that the miir method properly applies history to the	Test that the output can be pro- cessed back to an m-file.	pass
	plist(filename) constructor.	1) Check that the save method doesn't change the input ob- ject 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
15	Tests that the MIIR method properly applies history to the plist(conn) constructor.	Test that the output can be processed back with the rebuild method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'miir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
16	Tests that the MIIR method properly applies history to the plist(type) constructor.	Test that the output can be processed back with the rebuild method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'miir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
17	Tests that the MIIR method properly applies history to the plist(pzmodel) constructor.	Test that the output can be processed back with the rebuild method.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir			
		1) Check that the last entry in the history of 'out' corresponds to 'miir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
18	Tests that the MIIR method properly applies history to the plist(parfrac) constructor.	Test that the output can be processed back with the rebuild method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'miir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
19	Tests that the MIIR method properly applies history to the plist(<plist-object>)</plist-object>	Test that the output can be processed back with the rebuild method.	pass
	constructor.	1) Check that the last entry in the history of 'out' corresponds to 'miir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
20	Tests that the MIIR method properly applies history to the a,b constructor.	Test that the output can be processed back with the rebuild method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'miir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
21	Tests that the MIIR method properly applies history to the pole/zero model + plist object	Test that the output can be processed back with the rebuild method.	pass
	constructor.	1) Check that the last entry in the history of 'out' corresponds to 'miir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
22	Tests that the MIIR method properly applies history to the partial fraction model + plist	Test that the output can be processed back with the rebuild method.	pass
	object constructor.	1) Check that the last entry in the history of 'out' corresponds to 'miir'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
23	Tests that the MIIR method properly applies history to the conn+Id constructor.	Test that the output can be processed back with the rebuild method.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

miir/miir			
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'miir'. 2) Check that the 're-	
		build' method produces the same	
		object as 'out'.	
24	Tests that the MIIR method	Test that the output can be	pass
24	properly applies history to the a	processed back with the rebuild	
	+ b $+$ fs object constructor.	method.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'miir'. 2) Check that the 're-	
		build' method produces the same	
		object as 'out'.	

Table 208: Unit tests for miir/miir.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

miir/ne			
01	Tests that the getInfo call works for this method	Test that the getInfo call works	pass
	for this method.	individually	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the ne method works	Test that the ne method works	pass
	with a vector of MIIR objects as	for a vector of MIIR objects as	
	input.	input. Test the positive and the	
		negative case.	
		1) Check the output of the ne	pass
		function.	
03	Tests that the ne method works	Test that the ne method works	pass
	with a matrix of MIIR objects	for a matrix of MIIR objects as	
	as input.	input. lest the positive and the	
		1) Check the output of the ne	Daga
		function	pass
	Tests that the ne method works	The ne method doesn't works for	nage
04	with a list of MIIB objects as	a list of MIIB objects as input	pass
	input.	Nothing to do.	
	- mp au		pass
05	Tests that the ne method works	The ne method doesn't works for	pass
05	with a mix of different shaped	a list of MIIR objects as input.	
	MIIR objects as input.	Nothing to do.	
			pass
06	Tests that the ne method	The ne method doesn't change	pass
	properly applies history.	the MIIR object, thus will no his-	
		tory added. Nothing to do	
			pass
07	Test the ne method with an	lest the ne method with the ex-	pass
	exception list. The function	ception 'name'. Use the option	
	mir/ne use the function mir/eq	It is page any to add 'arouted' to	
	possibilities of the exception list	the exception list because iir is	
	possibilities of the exception list.	created at an other time	
		1) Check that each output con-	pass
		tains the correct data.	Pass
	Test the ne method with an	Test that the ne method uses the	pass
08	exception list which is in a plist.	exception list in a plist.	1
		1) Check that each output con-	pass
		tains the correct data.	

Table 209: Unit tests for miir/ne.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

miir/rebuild			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInto call re-	pass
		turned an minito object in all	
		cases. 2) Check that all plists	
	Tests that the rebuild method	Tast that the parameters.	
02	works with a water of MIP	rest that the rebuild method	pass
	works with a vector of MIIR	works for a vector of white ob-	
	objects as input.	1) Check the rebuilt output	nagg
	Tosts that the rebuild method	Tost that the rebuild method	pass
03	works with a matrix of MIIB	works for a matrix of MIR ob	pass
	objects as input	iects as input	
	objects as input.	1) Check the rebuilt output	pass
	Tests that the rebuild method	Test that the rebuild method	pass
04	works with a list of MIIR	works for a list of MIIR objects	Pass
	objects as input.	as input.	
	T T	1) Check the rebuilt output.	pass
05	Tests that the rebuild method	Test that the rebuild method	pass
05	works with a mix of different	works with an input of matrices	
	shaped MIIR objects as input.	and vectors and single MIIR ob-	
		jects.	
		1) Check the rebuilt output.	pass
06	Tests that the rebuild method	The method rebuild doesn't	pass
00	properly applies history.	change the data, thus it is not	
		possible to check the history.	
		Nothing to do.	
			pass
07	Check that the rebuild method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Uneck that the output con-	pass
		tains the right number of ob-	
		method produces the serve of	
		inethod produces the same ob-	
		ject as out.	

Table 210: Unit tests for miir/rebuild.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir/redesign			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually. 1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the redesign method works with a vector of MIIR objects as input.	Test that the redesign method works for a vector of MIIR ob- jects as input. To keep this UTP simple use for the vector only one filter object. The different filters will be tested in an other UTP. 1) Check that the number of el-	pass
		ements in 'out' is the same as in 'iirvec' 2) Check that each out- put MIIR contains the correct data.	
03	Tests that the redesign method works with a matrix of MIIR objects as input.	Test that the redesign method works for a matrix of MIIR ob- jects as input. To keep this UTP simple use for the matrix only one filter object. The different fil- ters will be tested in an other UTP.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'iirmat' 2) Check that each out- put MIIR contains the correct data.	pass
04	Tests that the redesign method works with a list of MIIR objects as input.	Test that the redesign method works for a list of MIIR objects as input. To keep this UTP sim- ple use for the list only one filter object. The different filters will be tested in an other UTP.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output MIIR contains the correct data.	pass
05	Tests that the redesign method works with a mix of different shaped MIIR objects as input.	Test that the redesign method works with an input of matrices and vectors and single MIIR ob- jects. To keep this UTP simple use for the vector only one filter object. The different filters will be tested in an other UTP.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir/redesign			
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		MIIR contains the correct data.	
06	Tests that the redesign method	Test that the result of applying	pass
00	properly applies history.	the redesign method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'redesign'. 2) Check that re-	
		built object is the same object as	
		the input.	
07	Tests that the redesign method	Tests that the redesign method	pass
07	redesigns a standard filter type.	redesigns a standard filter type.	
		1) Check the output 2) Check the	pass
		rebuilt object	
00	Tests that the redesign method	Tests that the redesign method	pass
08	redesigns a pzmodel filter type.	redesigns a pzmodel filter type.	
		1) Check the output 2) Check the	pass
		rebuilt object	
00	Tests that the redesign method	Tests that the redesign method	pass
09	redesigns a parfrac filter type.	redesigns a parfrac filter type.	
		1) Check the output 2) Check the	pass
		rebuilt object	

Table	211:	Unit	tests	for	miir	redesign.
					/	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

miir/resp			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the resp method works with a vector of MIIR objects as input.	Test that the resp method works for a vector of MIIR objects as input. Test the method with an output and with no output (a di- agram must appear)	pass
		1) Test the right number of lines in the diagram. 2) Check that the number of elements in 'out' is the same as in 'iirv' 3) Check that each output MIIR contains the correct data.	pass
03	Tests that the resp method works with a matrix of MIIR objects as input.	Test that the resp method works for a matrix of MIIR objects as input. Test the method with an output and with no output (a di- agram must appear)	pass
		1) Test the right number of lines in the diagram. 2) Check that the number of elements in 'out' is the same as in 'iirmat' 3) Check that each output MIIR contains the correct data.	pass
04	Tests that the resp method works with a list of MIIR objects as input.	Test that the resp method works for a list of MIIR objects as in- put. Test the method with an output and with no output (a di- agram must appear)	pass
		1) Test the right number of lines in the diagram. 2) Check that the number of elements in 'out' is the same as in 'iirmat' 3) Check that each output MIIR contains the correct data.	pass
05	Tests that the resp method works with a mix of different shaped MIIR objects as input.	Test that the resp method works with an input of matrices and vectors and single MIIR objects. Test the method with an output and with no output (a diagram must appear)	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

miir/resp			
		1) Test the right number of lines in the diagram. 2) Check that the number of elements in 'out' is the same as in 'iirmat' 3) Check that each output MIIR contains the correct data.	pass
06	properly applies history.	inst that the result of apply- ing the resp method can be pro- cessed back.1) Check that the last entry in the history of 'out' corresponds	pass
	Tosts that modify command	to 'resp'. 2) Check that re-built object is the same object as the input.	Dage
07	plots the response into a diagram.	plots the response into a dia- gram. 1) Check the response diagram.	pass
08	Test the shape of the output.	Test that the output AO of the resp method keeps the shape of the used input f vector.	pass
		data doesn't change.	pass
09	Check that the resp method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
10	Check that the resp method uses the x-data of an input AO for f-vector.	Call the method with different method to pass an AO in. 1) Check the output 2) Check that the 'rebuild' method pro-	pass pass
11	Check that the resp method uses the specified f-vector to compute the response.	duces the same object as 'out'.Call the method with differentmethod to pass an f-vector in.1) Check the output 2) Checkthat the 'rebuild' method pro-duces the same object as 'out'	pass pass
12	Check that the resp method uses the specified f1, f2, and nf to compute the response.	Call the method with different method to pass f1, f2, and nf in. 1) Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'	pass pass
13	Check that the resp method uses the specified f1, f2, and nf to compute the response.	Call the method with different method to pass f1, f2, and nf in.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

miir/resp			
		1) Check the output 2) Check	pass
		that the 'rebuild' method pro-	
		duces the same object as 'out'.	
14	Check that the resp method the	Check that the resp method the	pass
14	response of a serial filter bank.	response of a serial filter bank.	
		1) Check the output 2) Check	pass
		that the 'rebuild' method pro-	
		duces the same object as 'out'.	
15	Check that the resp method the	Check that the resp method the	pass
10	response of a parallel filter bank.	response of a parallel filter bank.	
		1) Check the output 2) Check	pass
		that the 'rebuild' method pro-	
		duces the same object as 'out'.	

Table 212: Unit tests for miir/resp. $% \left({{{\rm{Table}}}} \right) = {{{\rm{Table}}}} \right)$



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir/save			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the save method works with a vector of MIIR objects as input.	Test that the save method works for a vector of MIIR objects as input. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of ele- ments in 'out1' and 'out2' are the same as in 'iirv' 2) Check that the loaded objects are the same as the saved objects. 3) The out- puts 'out1' and 'out2' must be the same.	pass
03	Tests that the save method works with a matrix of MIIR objects as input.	Test that the save method works for a matrix of MIIR objects as input. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out1' and 'out2' are the same as in 'iirm' 2) Check that the loaded objects are the same as the saved objects. 3) The outputs 'out1' and 'out2' must be the same.	pass
04	Tests that the save method works with a list of MIIR objects as input.	Test that the save method works for a list of MIIR objects as in- put. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out1' and 'out2' are the same as in the list 2) Check that the loaded objects are the same as the saved objects. 3) The outputs 'out1' and 'out2' must be the same.	pass
05	Tests that the save method works with a mix of different shaped MIIR objects as input.	Test that the save method works with an input of matrices and vectors and single MIIR objects. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output MIIR object contains the correct data.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

miir/save			
06	Tests that the save method	Test that the result of apply-	pass
	properly applies history.	ing the save method can be pro-	
		cessed back to an m-file. Do this	
		for both extensions 'mat' and	
		'xml'	
		1) Check that the history applies	pass
		to the output object. Check that	
		save doesn't add a history step to	
		the input object. 2) Check that	
		the read object doesn't contain	
		the save + load history steps.	
		3) Check that the method re-	
		build produces the same object	
		as 'out'.	
07	Tests that the save method	Use the save method with the	pass
	works with the modify	modifier command.	
	command.	1) Check that the save method	pass
		doesn't apply the history. 2)	
		Check the output against the in-	
		put.	
08	Control the method with a plist	Test that the save method uses	pass
	Control the method with a plist.	the filename which is stored in a	
		plist.	
		1) Check the output	pass
09	Test the save method with	Save all standard MIIR objects	pass
05	standard MIIR objects.	with both extensions.	
		1) Check the output	pass
10	Test the save method with	Save MIIR object which is cre-	pass
10	MIIR object which is created	ated from a pzmodel.	
	from a pole/zero model	1) Check the output	pass

Table 213: Unit tests for miir/save.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

miir/setHistin			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.9	Tests that the setHistin method	The setHistin method is not de-	pass
02	works with a vector of MIIR	signed for vectors, for that reason	
	objects as input.	must this call fail.	
		1) Nothing to check	pass
0.2	Tests that the setHistin method	The setHistin method is not de-	pass
03	works with a matrix of MIIR	signed for matrices, for that rea-	
	objects as input.	son must this call fail.	
		1) Nothing to check	pass
0.4	Tests that the setHistin method	The setHistin method is not de-	pass
04	works with a list of MIIR	signed for input lists, for that	1
	objects as input.	reason must this call fail.	
		1) Nothing to check	pass
	Tests that the setHistin method	The setHistin method is not de-	pass
05	works with a mix of different	signed for this call. for that rea-	1
	shaped MIIR objects as input.	son must this call fail.	
		1) Nothing to check	pass
	Tests that the setHistin method	Test that the result of applying	pass
06	properly applies history and	the setHistin method can be pro-	P
	that the option 'internal'	cessed back to an m-file.	
	suppresses the history.	1) Check that the last entry in	pass
		the history of 'out1' corresponds	P
		to 'setHistin' 2) Check that the	
		last entry in the history of 'out?'	
		NOT corresponds to 'setHistin'	
		3) Check that the method re-	
		build produces the same object	
		as 'out'	
	Tests that the setHistin method	Test that the setHistin method	nass
07	can modify the input MIB	can modify the input MIR ob-	pass
	object	ight hours the input with no output	
	object.	1) Check that 'iirbp' and 'aip' are	paga
		now different 2) Check that 'sin'	pass
		how different. 2) Check that all	
	Tosts that the set Histin method	Tost that the set Uistin method	negg
08	iests that the set instin method	iest that the set instin method	pass
	can set the property with a	with a value in a relief	
	pust.	with a value in a plist.	
		1) Uneck that 'ain' has the cor-	pass
		rect nistin neid 2) Check that	
		the method rebuild produces the	
		same object as 'out'.	

Table 214: Unit tests for miir/setHistin.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

miir/setHistout			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.0	Tests that the setHistout	The setHistout should set the	pass
02	method works with a vector of	output history (histout) of each	-
	MIIR objects as input.	input.	
	0 1	1) Check the histout has the cor-	pass
		rect values	
	Tests that the setHistout	The setHistout should set the	pass
03	method works with a matrix of	output history (histout) of each	1
	MIIR objects as input.	input.	
	0 I	1) Check the histout has the cor-	pass
		rect values	
	Tests that the setHistout	The setHistout should set the	pass
04	method works with a list of	output history (histout) of each	1
	MIIR objects as input.	input.	
	0 I	1) Check the histout has the cor-	pass
		rect values	
	Tests that the setHistout	The setHistout method is not de-	pass
05	method works with a mix of	signed for this call, for that rea-	P
	different shaped MIIR objects	son must this call fail.	
	as input.	1) Check the histout has the cor-	pass
	I m	rect values	1
	Tests that the setHistout	Test that the result of apply-	pass
06	method properly applies history	ing the setHistout method can be	1
	and that the option 'internal'	processed back to an m-file.	
	suppresses the history.	1) Check that the last entry	pass
		in the history of 'out1' corre-	
		sponds to 'setHistout'. 2) Check	
		that the last entry in the his-	
		tory of 'out2' NOT corresponds	
		to 'setHistout'. 3) Check that	
		the method rebuild produces the	
		same object as 'out'.	
	Tests that the setHistout	Test that the setHistout method	pass
07	method can modify the input	can modify the input MIR ob-	1
	MIIR object.	ject by calling with no output.	
		1) Check that 'iirhp' and 'ain' are	pass
		now different. 2) Check that 'ain'	1
		has the correct histout field	
0.0	Tests that the setHistout	Test that the setHistout method	pass
08	method can set the property	can modify the property 'histout'	1
	with a plist.	with a value in a plist.	
1	1 1	1 I I I I I I I I I I I I I I I I I I I	1



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir/setHistout		
	1) Check that 'ain' has the cor-	pass
	rect histout field 2) Check that	
	the method rebuild produces the	
	same object as 'out'.	

Table 215: Unit tests for miir/setHistout.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir/setIunits			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases 2) Check that all plicts	pass
		have the correct parameters.	
02	Tests that the setIunits method works with a vector of MIIR	Test that the setIunits method works for a vector of MIIR ob-	pass
	objects as input.	 jects as input. 1) Check that the number of elements in 'out' is the same as in 'iiry' 2) Check that each output. 	pass
		contains the correct data.	
03	Tests that the setIunits method works with a matrix of MIIR objects as input.	Test that the setIunits method works for a matrix of MIIR ob- jects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'iirm' 2) Check that each output contains the correct data.	pass
04	Tests that the setIunits method works with a list of MIIR objects as input.	Test that the setIunits method works for a list of MIIR objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the setIunits method works with a mix of different shaped MIIR objects as input.	Test that the setIunits method works with an input of matrices and vectors and single MIIR ob- jects.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the setIunits method properly applies history and that the option 'internal'	Test that the result of applying the setIunits method can be pro- cessed back to an m-file.	pass
	suppresses the history.	 Check that the last entry in the history of 'out1' corresponds to 'setIunits'. 2) Check that the last entry in the history of 'out2' NOT corresponds to 'setIunits'. Check that the method re- build produces the same object as 'out'. 	pass
07	can modify the input MIIR object.	can modify the input MIIR object by calling with no output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

miir/setIunits			
		1) Check that 'iirhp' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct iunits field	
08	Tests that the setIunits method	Test that the setIunits method	pass
00	can set the property with a	can modify the property 'iunits'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect iunits field 2) Check that	
		the method rebuild produces the	
		same object as 'out'.	
00	Check that the setIunits method	Call the method with a list of	pass
09	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 216: Unit tests for miir/setIunits.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



, ,			
Tests that	the getInfo call works	Test that the getInfo call works	pass
for this me	thod.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
Tests that	the setName method	Test that the setName method	pass
works with	a vector of MIIR	works for a vector of MIIR ob-	
objects as i	input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'iirv' 2) Check that each output	
		contains the correct data.	
Tests that	the setName method	Test that the setName method	pass
works with	a matrix of MIIR	works for a matrix of MIIR ob-	
objects as i	input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'iirm' 2) Check that each output	
		contains the correct data.	
Tests that	the setName method	Test that the setName method	pass
04 works with	a list of MIIR	works for a list of MIIR objects	
objects as i	input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		contains the correct data.	
Tests that	the setName method	Test that the setName method	pass
works with	a mix of different	works with an input of matrices	
shaped MI	IR objects as input.	and vectors and single MIIR ob-	
	· -	jects.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		contains the correct data.	
Tests that	the setName method	Test that the result of applying	pass
properly ap	oplies history and	the setName method can be pro-	
that the op	otion 'internal'	cessed back to an m-file.	
suppresses	the history.	1) Check that the last entry in	pass
		the history of 'out1' corresponds	
		to 'setName'. 2) Check that the	
		last entry in the history of 'out2'	
		NOT corresponds to 'setName'.	
		3) Check that the method re-	
		build produces the same object	
		as 'out'.	
Tests that	the setName method	Test that the setName method	pass
can modify	the input MIIR	can modify the input MIIR ob-	
	-	i oot he colling with no output	


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

miir/setName			
		1) Check that 'iirhp' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct name field	
08	Tests that the setName method	Test that the setName method	pass
00	can set the property with a	can modify the property 'name'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect name field 2) Check that	
		the method rebuild produces the	
		same object as 'out'.	
00	Check that the setName method	Call the method with a list of	pass
09	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 217: Unit tests for miir/setName.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



miir/setOunits			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the setOunits method	Test that the setOunits method	pass
02	works with a vector of MIIR	works for a vector of MIIR ob-	
	objects as input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'iirv' 2) Check that each output	
		contains the correct data.	
03	Tests that the setOunits method	Test that the setOunits method	pass
05	works with a matrix of MIIR	works for a matrix of MIIR ob-	
	objects as input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'iirm' 2) Check that each output	
		contains the correct data.	
04	Tests that the setOunits method	Test that the setOunits method	pass
04	works with a list of MIIR	works for a list of MIIR objects	
	objects as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		contains the correct data.	
05	Tests that the setOunits method	Test that the setOunits method	pass
00	works with a mix of different	works with an input of matrices	
	shaped MIIR objects as input.	and vectors and single MIIR ob-	
		jects.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		contains the correct data.	
06	Tests that the setOunits method	Test that the result of apply-	pass
	properly applies history and	ing the setOunits method can be	
	that the option 'internal'	processed back to an m-file.	
	suppresses the history.	1) Check that the last entry in	pass
		the history of 'out1' corresponds	
		to 'setOunits'. 2) Check that the	
		last entry in the history of 'out2'	
		NOT corresponds to 'setOunits'.	
		3) Check that the method re-	
		build produces the same object	
		as 'out'.	
07	Tests that the setOunits method	Test that the setOunits method	pass
	can modify the input MIIR	can modify the input MIIR ob-	
	object.	ject by calling with no output.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

miir/setOunits			
		1) Check that 'iirhp' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct ounits field	
08	Tests that the setOunits method	Test that the setOunits method	pass
00	can set the property with a	can modify the property 'ounits'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect ounits field 2) Check that	
		the method rebuild produces the	
		same object as 'out'.	
00	Check that the setOunits	Call the method with a list of	pass
09	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 218: Unit tests for miir/setOunits.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

miir/string			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the string method works with a vector of MIIR objects as input.	Test that the string method works for a vector of MIIR ob- jects as input.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
03	Tests that the string method works with a matrix of MIIR objects as input.	Test that the string method works for a matrix of MIIR ob- jects as input.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
04	Tests that the string method works with a list of MIIR objects as input.	Test that the string method works for a list of MIIR objects as input.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
05	Tests that the string method works with a mix of different shaped MIIR objects as input.	Test that the string method works with an input of matrices and vectors and single MIIR ob- jects.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
06	Tests that the string method properly applies history.	The method string doesn't change the data, thus it is not possible to check the history. Nothing to do.	pass
	Tests that the string method	The method string throws an or	pass
07	doesn't work if the MIIR object have more than one history step.	ror because the input object have more than one history step.	pass
			pass

Table 219: Unit tests for miir/string.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

miir/type			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	individually	
		1) Check that got Info call ro	nase
		turned an minfo object in all	pass
		cases 2) Check that all plists	
		have the correct parameters	
	Tosts that the type method	Tost that the type method works	nase
02	works with a vector of MIIB	for a vector of MIIB objects as	pass
	objects as input	input	
	objects as input.	1) Check the rebuilt output	nass
	Tests that the type method	Test that the type method works	pass
03	works with a matrix of MIIB	for a matrix of MIIB objects as	pass
	objects as input	input	
		1) Check the rebuilt output.	pass
0.4	Tests that the type method	Test that the type method works	pass
04	works with a list of MIIR	for a list of MIIR objects as in-	1
	objects as input.	put.	
		1) Check the rebuilt output.	pass
05	Tests that the type method	Test that the type method works	pass
60	works with a mix of different	with an input of matrices and	
	shaped MIIR objects as input.	vectors and single MIIR objects.	
		1) Check the rebuilt output.	pass
06	Tests that the type method	The method type doesn't change	pass
00	properly applies history.	the data, thus it is not possible	
		to check the history. Nothing to	
		do.	
			pass

Table 220: Unit tests for miir/type.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

parfrac/char			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the char methodworks with a vector ofPARFRAC objects as input.	Test that the char method works for a vector of PARFRAC ob- jects as input.	pass
		1) Check that the output contain at least each object name	pass
03	Tests that the char method works with a matrix of PARFRAC objects as input.	Test that the char method works for a matrix of PARFRAC ob- jects as input.	pass
		1) Check that the output contain at least each object name	pass
04	Tests that the char method works with a list of PARFRAC objects as input.	Test that the char method works for a list of PARFRAC objects as input.	pass
		1) Check that the output contain at least each object name	pass
05	Tests that the char method works with a mix of different shaped PARFRAC objects as input.	Test that the char method works with an input of matrices and vectors and single PARFRAC objects.	pass
		1) Check that the output contain at least each object name	pass
06	Tests that the char method properly applies history.	The method char doesn't change the data, thus it is not possible to check the history. Nothing to do.	pass
			pass

Table 221: Unit tests for parfrac/char.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

parfrac/copy			
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	

Table 222: Unit tests for parfrac/copy.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



parfrac/created			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters	pass
02	Tests that the created method works with a vector of PARFRAC objects as input.	Test that the created method works for a vector of PARFRAC objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'pfv' 2) Check that each output contains the correct data.	pass
03	Tests that the created method works with a matrix of PARFRAC objects as input.	Test that the created method works for a matrix of PARFRAC objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'pfm' 2) Check that each output contains the correct data.	pass
04	Tests that the created method works with a list of PARFRAC objects as input.	Test that the created method works for a list of PARFRAC ob- jects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the created method works with a mix of different shaped PARFRAC objects as input.	Test that the created method works with an input of matri- ces and vectors and single PAR- FRAC objects.	pass
	- Factoria (1997)	1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the created method properly applies history	This method doesn't change the input object, thus no history is added to the object.	pass
07	Tests that the created method can be used with the modify command.	Tests that the created method can be used with the modify command. 1) Check the single object 2)	pass pass pass
08	Tests that the created method retruns always a well defined time object even for an empty input object.	Test that the created method with an empty 'PARFRAC object	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



parfrac/created		
	1) Check that the output is a	pass
	time object with a ell defined	
	time.	

Table 223: Unit tests for parfrac/created.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



partrac/creator			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
U1	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the creator method	Test that the creator method	pass
02	works with a vector of	works for a vector of PARFRAC	
	PARFRAC objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'pfv' 2) Check that each output	
		contains the correct data.	
	Tests that the creator method	Test that the creator method	pass
03	works with a matrix of	works for a matrix of PARFRAC	1
	PARFRAC objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	P
		'pfm' 2) Check that each output	
		contains the correct data.	
	Tests that the creator method	The creator method doesn't work	nass
04	works with a list of PAREBAC	for a list of PARERAC objects as	pass
	objects as input	input	
	objects as input.	1) Nothing to test	nass
	Tests that the creater method	The creater method doesn't work	pass
05	works with a mix of different	with different shaped input of	pass
	shaped PARERAC objects as	iocts	
	input	1) Nothing to tost	neg
	Trata that the exector method	This method deser't sharper the	pass
06	Tests that the creator method	This method doesn't change the	pass
	property applies instory	input object, thus no mistory is	
		added to the object.	
		1) Nothing to check.	pass
07	lests that the creator method	lests that the creator method	pass
	can be used with the modify	can be used with the modify	
	command.	command.	
		1) Check the single object 2)	pass
		Check the matrix object	
08	Tests that the creator method	Test that the creator method	pass
	retruns all	uses the option 'all' direct or in	
	creator(s)/modifier(s) which are	a plist. The test file must have	
	in the history.	the modifier 'first', 'second' and	
		'third'	
		1) Check that out1 contains only	pass
		one creator 2) Check that out2	
		contain more creator/modifier	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



parfrac/creator			
09	Tests the negative case for the option 'all'.	Test that the creator method throws an error if the option 'all' is used in connection with a matrix/vector of PARFRAC ob- jects.	pass
		1) Nothing to test.	pass

Table 224: Unit tests for parfrac/creator.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

parfrac/display			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set in dividually	pass
		1) Check that getInfo call re- turned an minfo object in all cases 2) Check that all plists	pass
		have the correct parameters.	
02	Tests that the display method works with a vector of PARFRAC objects as input.	Test that the display method works for a vector of PARFRAC objects as input.	pass
		1) Check that the output contain at least each object name	pass
03	Tests that the display method works with a matrix of PARFRAC objects as input.	Test that the display method works for a matrix of PARFRAC objects as input.	pass
		1) Check that the output contain at least each object name	pass
04	Tests that the display method works with a list of PARFRAC objects as input.	Test that the display method works for a list of PARFRAC ob- jects as input.	pass
		1) Check that the output contain at least each object name	pass
05	Tests that the display method works with a mix of different shaped PARFRAC objects as input.	Test that the display method works with an input of matri- ces and vectors and single PAR- FRAC objects as.	pass
		1) Check that the output contain at least each object name	pass
06	Tests that the display method properly applies history.	The method display doesn't change the data, thus it is not possible to check the history. Nothing to do.	pass
			Pass

Table 225: Unit tests for parfrac/display.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

parfrac/eq			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the eq method works with a vector of PARFRAC objects as input.	Test that the eq method works for a vector of PARFRAC ob- jects as input. Test the positive and the negative case.	pass
		function.	pass
03	Tests that the eq method works with a matrix of PARFRAC objects as input.	Test that the eq method works for a matrix of PARFRAC ob- jects as input. Test the positive and the negative case.	pass
		1) Check the output of the eq function.	pass
04	Tests that the eq method works with a list of PARFRAC objects as input.	The eq method doesn't works for a list of PARFRAC objects as in- put. Nothing to do.	pass
			pass
05	Tests that the eq method works with a mix of different shaped PARFRAC objects as input.	The eq method doesn't works for a list of PARFRAC objects as in- put. Nothing to do.	pass
			pass
06	Tests that the eq method properly applies history.	The eq method doesn't change the PARFRAC object, thus will no history added. Nothing to do	pass
			pass
07	Test the eq method with an exception list. With the LTPDA toolbox 2.0 it is only possible to test the exception list with properties where a public set method exist.	Test the eq method with the exception 'name'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because 'pa' is created at an other time. 1) Check the output.	pass
0.0	Test the eq method with an	Test the eq method with the ex-	pass
08	exception list. With the LTPDA toolbox 2.0 it is only possible to test the exception list with properties where a public set method exist.	ception 'iunits'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because 'pa' is created at an other time. 1) Check the output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

parfrac/eq			
00	Test the eq method with an	Test the eq method with the ex-	pass
	exception list. With the LTPDA	ception 'ounits'. Use the option	
	toolbox 2.0 it is only possible to	'internal' to suppress the history.	
	test the exception list with	It is necessary to add 'created' to	
	properties where a public set	the exception list because 'pa' is	
	method exist.	created at an other time.	
		1) Check the output.	pass
10	Test the eq method with an	Test that the eq method uses the	pass
10	exception list which is in a plist.	exception list in a plist.	
		1) Check the output.	pass

Table 226: Unit tests for parfrac/eq.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

parfrac/get			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests the get method of the	Test that the get returns returns	pass
02	parfrac class.	the value of the specified prop-	
		erty. Do this for all properties of	
		the PARFRAC object.	
		1) Check the correct value of the	pass
		output	
02	Tests that the get method works	Test that the get returns returns	pass
05	with a plist.	the value of the specified prop-	
		erty which is defined in a plist.	
		1) Check the correct value of the	pass
		output	
04	Tests the get method of the	Test that the get throws an error	pass
04	parfrac class.	if the input are more than one	
		PARFRAC object.	
		1) Nothing to test	pass

Table 227: Unit tests for parfrac/get.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

parfrac/getlowerF	req		
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests the getlowerFreq method	Test that the getlowerFreq re-	pass
02	of the parfrac class.	turns the lowest frequence of the	
		pole in the parfrac object.	
		1) Check the output	pass
02	Tests the getlowerFreq method	Test that the getlowerFreq	pass
03	of the parfrac class.	throws an error if the input are	
		more than one parfrac.	
		1) Nothing to test	pass

Table 228: Unit tests for parfrac/getlowerFreq.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

parfrac/getupperH	req		
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests the getupperFreq method	Test that the getupperFreq re-	pass
02	of the parfrac class.	turns the lowest frequence of the	
		pole in the parfrac object.	
		1) Check the output	pass
02	Tests the getupperFreq method	Test that the getupperFreq	pass
03	of the parfrac class.	throws an error if the input are	
		more than one parfrac.	
		1) Nothing to test	pass

Table 229: Unit tests for parfrac/getupperFreq.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

parfrac/index			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameter.	pass
	Togta that the index method	Tagt that the index method	
02	rests that the mdex method	werks for a vector of DADEDAC	pass
	DAPEDAC objects of input	works for a vector of FARFRAC	
	I AIT ITAC Objects as input.	indexing should work: $I = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$	
		$\begin{bmatrix} \text{Indexing should work, I} = \begin{bmatrix} 12.5 \\ 1.2 \end{bmatrix}$	
		1) Check that the index method	nass
		selects the correct object.	pass
	Tests that the index method	Test that the index method	pass
03	works with a matrix of	works for a matrix of PARFRAC	1
	PARFRAC objects as input.	objects as input. The following	
	0 1	indexing should work: $I = [135]$	
] or $(I/J) = [(1,1), (1,2), (1,3)]$	
		2 4 6] [(2,1), (2,2), (2,3)]	
		1) Check that the index method	pass
		selects the correct object.	
04	Tests that the index method	The index method doesn't work	pass
04	works with a list of PARFRAC	for a list of PARFRAC objects as	
	objects as input.	input.	
		1) Nothing to test.	pass
05	Tests that the index method	Test that the result of index have	pass
	properly applies history.	an additional history step.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
	Tests that the index method	To index.	
06	works for the modifier	works for the modifier command	pass
	command	1) Check that the history-plist	nass
	commune.	contains the used indices. 2)	pass
		Check that the index method se-	
		lects the correct object	
07		Test that the index method can	pass
07	Control the method with a plist.	be controled with a plist.	-
		1) Check that the history-plist	pass
		contains the used indices. 2)	
		Check that the index method se-	
		lects the correct object	
08	Test that the index method	Test that the index method se-	pass
	selects more objects if I have	lects more objects if I have more	
	more indices.	indices.	
		1) Check that the history-plist	pass
		contains the used indices. 2)	
		Check that the index method se-	
		lects the correct object	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



parfrac	/index
· /	

Table 230: Unit tests for parfrac/index.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



parfrac/isprop			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists	pass
		have the correct parameters.	
02	Tests that the isprop method works with a vector of PARFRAC objects as input.	Test that the isprop method works for a vector of PARFRAC objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'pfv' 2) Check that each output contains the correct data.	pass
03	Tests that the isprop method works with a matrix of PARFRAC objects as input.	Test that the isprop method works for a matrix of PARFRAC objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'pfm' 2) Check that each output contains the correct data.	pass
04	Tests that the isprop method works with a list of PARFRAC objects as input.	Test that the isprop method works for a list of PARFRAC ob- jects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the isprop method works with a mix of different shaped PARFRAC objects as input.	Test that the isprop method works with an input of matri- ces and vectors and single PAR- FRAC objects.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the isprop method properly applies history.	The method isprop doesn't change the object, thus it is not necessary to apply history.	pass
	Tests that the isnron method	Test that the isnron method	pass
07	works for each property.	works for the properties: 'res', 'poles', 'pmul', 'dir', 'iunits', 'ounits', 'hist', 'name'	pass
		1) Check that each output con-	pass
08	Test the negative case and the	Test that the isprop method	pass
00	not function command.	retrun false for a unknown prop- erty and for methods of the ob- ject.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



parfrac/isprop			
	1) Check	that each output con-	pass
	tains the c	correct data.	

Table 231: Unit tests for parfrac/isprop.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



parfrac/loadobj			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check the shape of the loaded	pass
		objects.	

Table 232: Unit tests for parfrac/loadobj.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

parfrac/ne			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually. 1) Check that getInfo call re-	pass
		turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	
02	Tests that the ne method works with a vector of PARFRAC objects as input.	Test that the ne method works for a vector of PARFRAC ob- jects as input. Test the positive and the negative case.	pass
		1) Check the output of the ne function.	pass
03	Tests that the ne method works with a matrix of PARFRAC objects as input.	Test that the ne method works for a matrix of PARFRAC ob- jects as input. Test the positive and the negative case.	pass
		1) Check the output of the ne function.	pass
04	Tests that the ne method works with a list of PARFRAC objects as input.	The ne method doesn't works for a list of PARFRAC objects as in- put. Nothing to do.	pass
			pass
05	Tests that the ne method works with a mix of different shaped PARFRAC objects as input.	The ne method doesn't works for a list of PARFRAC objects as in- put. Nothing to do.	pass
			pass
06	Tests that the ne method properly applies history.	The ne method doesn't change the PARFRAC object, thus will no history added. Nothing to do	pass
	Tract the mean that are the second	The static second state of second states and	pass
07	exception list. The function parfrac/ne use the function parfrac/eq so it is not necessary to check all possibilities of the exception list.	Test the ne method with the exception 'name'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because pf is created at an other time.	pass
		1) Oneck that each output con-	pass
08	Test the ne method with an exception list which is in a plist	Test that the ne method uses the exception list in a plist	pass
	exception list which is in a plist.	1) Check that each output con- tains the correct data.	pass

Table 233: Unit tests for parfrac/ne.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

parfrac/parfrac			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists	pass
		have the correct parameters.	
02	Tests that the parfrac method works with a vector of PARFRAC objects as input.	Test that the parfrac method works with a vector of PAR- FRAC objects as input.	pass
		 Check that the shape of the output PARFRACs is the same as the input shape. Check that each output PARFRAC is a copy of the input PARFRAC. Check that the copy have an additional history step. 	pass
03	Tests that the parfrac method works with a matrix of PARFRAC objects as input.	Test that the parfrac method works with a matrix of PAR- FRAC objects as input.	pass
		 Check that the snape of the output PARFRACs is the same as the input shape. Check that each output PARFRAC is a copy of the input PARFRAC. Check that the copy have an additional history step. 	pass
04	Tests that the parfrac method works with a list of PARFRAC objects as input.	Test that the parfrac method works with a list of PARFRAC objects as input.	pass
		 Check that the number of elements in 'out' is the same of the number in the input. 2) Check that each output PARFRAC is a copy of the input PARFRAC. Check that the copy have an additional history step. 	pass
05	Tests that the parfrac method works with a mix of different shaped PARFRACs as input.	Test that the parfrac method works with a mix of different shaped PARFRACs as input.	pass
		 Check that the number of elements in 'out' is the same of the number in the input. 2) Check that each output PARFRAC is a copy of the input PARFRAC. Check that the copy have an additional history step. 	pass
06	Tests that the parfrac method properly applies history.	Test that the result of applying the parfrac method can be pro- cessed back.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

parfrac/parfrac			
		1) Check that the last entry in the history of 'out' corre- sponds to 'parfrac'. 2) Check that the method rebuild pro- duces the same object as 'out'.	pass
07	properly applies history to the copy constructor.	rest that the output can be pro- cessed back with the 'rebuild' method. Test the constructor with a different number of in- puts.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'parfrac'. 2) Check that the original objects are not changed by the setter function 3) Check that the method rebuild pro- duces the same object as 'out'.	pass
08	Tests that the parfrac method properly applies history to the read MAT-file constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the loaded object is the same as the saved object. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
09	Tests that the parfrac method properly applies history to the read XML-file constructor.	Test that the output can be pro- cessed back with the 'rebuild' method. 1) Check that the loaded ob- ject is the same as the saved ob-	pass pass
		ject. 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	
10	Tests that the parfrac method properly doesn't apply history to the struct constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'parfrac'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
11	Tests that the parfrac method properly applies history to the rational constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'parfrac'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



parfrac/parfrac			
12	Tests that the parfrac method properly applies history to the pzmodel constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'parfrac'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
13	Tests that the parfrac method properly applies history to the	Test that the output can be pro- cessed back to an m-file.	pass
	plist(filename) constructor.	1) Check that the save method doesn't change the input ob- ject 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
14	Tests that the PARFRAC method properly applies history to the plist(conn) constructor.	Test that the output can be processed back with the rebuild method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'parfrac'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
15	Tests that the PARFRAC method properly applies history to the plist(res poles dir)	Test that the output can be processed back with the rebuild method.	pass
	constructor.	1) Check that the last entry in the history of 'out' corresponds to 'parfrac'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
16	Tests that the PARFRAC method properly applies history to the plist(pzmodel)	Test that the output can be processed back with the rebuild method.	pass
	constructor.	1) Check that the last entry in the history of 'out' corresponds to 'parfrac'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
17	Tests that the PARFRAC method properly applies history to the plist(parfrac) constructor.	Test that the output can be processed back with the rebuild method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'parfrac'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
18	Tests that the PARFRAC method properly applies history to the plist(<plist-object>) constructor.</plist-object>	Test that the output can be processed back with the rebuild method.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

parfrac/parfrac			
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'parfrac'. 2) Check that the	
		'rebuild' method produces the	
		same object as 'out'.	
10	Tests that the PARFRAC	Test that the output can be	pass
19	method properly applies history	processed back with the rebuild	
	to the conn+Id constructor.	method.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'parfrac'. 2) Check that the	
		'rebuild' method produces the	
		same object as 'out'.	
20	Tests that the PARFRAC	Test that the output can be	pass
20	method properly applies history	processed back with the rebuild	
	to the res $+$ poles $+$ direct	method.	
	terms object constructor.	1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'parfrac'. 2) Check that the	
		'rebuild' method produces the	
		same object as 'out'.	
21	Tests that the PARFRAC	Test that the output can be	pass
21	method properly applies history	processed back with the rebuild	
	to the res $+$ poles $+$ direct	method.	
	terms + name object	1) Check that the last entry in	pass
	constructor.	the history of 'out' corresponds	
		to 'parfrac'. 2) Check that the	
		'rebuild' method produces the	
		same object as 'out'.	
22	Tests that the PARFRAC	Test that the output can be	pass
	method properly applies history	processed back with the rebuild	
	to the res $+$ poles $+$ direct	method.	
	terms + name + iunits + ounits	1) Check that the last entry in	pass
	object constructor.	the history of 'out' corresponds	
		to 'parfrac'. 2) Check that the	
		'rebuild' method produces the	
		same object as 'out'.	

Table 234: Unit tests for parfrac/parfrac.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

parfrac/rebuild			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the rebuild method	Test that the rebuild method	pass
02	works with a vector of	works for a vector of PARFRAC	
	PARFRAC objects as input.	objects as input.	
		1) Check the rebuilt output.	pass
03	Tests that the rebuild method	Test that the rebuild method	pass
00	works with a matrix of	works for a matrix of PARFRAC	
	PARFRAC objects as input.	objects as input.	
		1) Check the rebuilt output.	pass
04	Tests that the rebuild method	Test that the rebuild method	pass
01	works with a list of PARFRAC	works for a list of PARFRAC ob-	
	objects as input.	jects as input.	
		1) Check the rebuilt output.	pass
05	Tests that the rebuild method	Test that the rebuild method	pass
	works with a mix of different	works with an input of matri-	
	shaped PARFRAC objects as	ces and vectors and single PAR-	
	input.	FRAC objects.	
		1) Check the rebuilt output.	pass
06	Tests that the rebuild method	The method rebuild doesn't	pass
	properly applies history.	change the data, thus it is not	
		possible to check the history.	
		Nothing to do.	
			pass
07	Check that the rebuild method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 235: Unit tests for parfrac/rebuild.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



parfrac/resp			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.0	Tests that the resp method	Test that the resp method works	pass
02	works with a vector of	for a vector of PARFRAC ob-	-
	PARFRAC objects as input.	jects as input. Test the method	
		with an output and with no out-	
		put (a diagram must appear)	
		1) Test the right number of lines	pass
		in the diagram. 2) Check that the	1
		number of elements in 'out' is the	
		same as in 'pfy' 3) Check that	
		each output PAREBAC contains	
		the correct data	
	Tests that the resp method	Tests that the resp method works	nass
03	works with a matrix of	with a matrix of PARERAC ob-	pass
	PARERAC objects as input	jects as input. Test the method	
	I ARTITAC Objects as input.	with an output and with no out	
		put (a diagram must appear)	
		1) Test the right number of lines	Daga
		in the diagram 2) Check that the	pass
		number of elements in 'out' is the	
		same as in 'nfm' 2) Check that	
		same as in pin 5) Check that	
		the service date	
		the correct data.	
04	lests that the resp method	Lests that the resp method works	pass
	works with a list of PARFRAC	with a list of PARFRAC objects	
	objects as input.	as input. Test the method with	
		an output and with no output (a	
		diagram must appear)	
		1) Test the right number of lines	pass
		in the diagram. 2) Check that the	
		number of elements in 'out' is the	
		same as in 'rain' 3) Check that	
		each output PARFRAC contains	
		the correct data.	
05	Tests that the resp method	Tests that the resp method works	pass
	works with a mix of different	with a mix of different shaped	
	shaped PARFRAC objects as	PARFRAC objects as input.	
	input.	Test the method with an output	
		and with no output (a diagram	
		must appear)	
1	I Contraction of the second	,,	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

parfrac/resp			
		1) Test the right number of lines in the diagram. 2) Check that the number of elements in 'out' is the same as in 'rain' 3) Check that each output PARFRAC contains the correct data.	pass
06	properly applies history.	Test that the result of apply- ing the resp method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'resp'. 2) Check that re-built object is the same object as the input.	pass
07	Tests that modify command plots the response into a diagram.	Tests that modify command plots the response into a dia- gram.	pass
08	Test the shape of the output.	Test that the output AO of the resp method keeps the shape of the used input f vector	pass
		 the used input i vector. Check that the shape of the data doesn't change. 	pass
09	Check that the resp method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
10	Check that the resp method uses the x-data of an input AO for f-vector.	Call the method with different method to pass an AO in. 1) Check the output 2) Check that the 'rebuild' method pro-	pass pass
11	Check that the resp method uses the specified f-vector to compute the response.	duces the same object as 'out'.Call the method with differentmethod to pass an f-vector in.1) Check the output 2) Check	pass
10	Check that the resp method	that the 'rebuild' method pro- duces the same object as 'out'. Call the method with different	pass
12	uses the specified f1, f2, and nf to compute the response.	method to pass f1, f2, and nf in. 1) Check the output 2) Check that the 'rebuild' method pro-	pass
13	Check that the resp method uses the specified f1, f2, and nf to compute the response.	duces the same object as 'out'. Call the method with different method to pass f1, f2, and nf in.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



parfrac/resp		
	1) Check the output 2) Check	pass
	that the 'rebuild' method pro-	
	duces the same object as 'out'.	

Table 236: Unit tests for parfrac/resp.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

parfrac/save			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the save method works with a vector of PARFRAC objects as input.	Test that the save method works for a vector of PARFRAC ob- jects as input. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out1' and 'out2' are the same as in 'pfv' 2) Check that the loaded objects are the same as the saved objects. 3) The out- puts 'out1' and 'out2' must be the same.	pass
03	Tests that the save method works with a matrix of PARFRAC objects as input.	Test that the save method works for a matrix of PARFRAC ob- jects as input. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of ele- ments in 'out1' and 'out2' are the same as in 'pfm' 2) Check that the loaded objects are the same as the saved objects. 3) The out- puts 'out1' and 'out2' must be the same.	pass
04	Tests that the save method works with a list of PARFRAC objects as input.	Test that the save method works for a list of PARFRAC objects as input. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out1' and 'out2' are the same as in the list 2) Check that the loaded objects are the same as the saved objects. 3) The outputs 'out1' and 'out2' must be the same.	pass
05	Tests that the save method works with a mix of different shaped PARFRAC objects as input.	Test that the save method works with an input of matrices and vectors and single PARFRAC objects. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each out- put PARFRAC object contains the correct data.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

parfrac/save			
06	Tests that the save method	Test that the result of apply-	pass
00	properly applies history.	ing the save method can be pro-	
		cessed back to an m-file. Do this	
		for both extensions 'mat' and	
		'xml'	
		1) Check that the history applies	pass
		to the output object. Check that	
		save doesn't add a history step to	
		the input object. 2) Check that	
		the read object doesn't contain	
		the save + load history steps.	
		3) Check that the method re-	
		build produces the same object	
		as 'out'.	
07	Tests that the save method	Use the save method with the	pass
07	works with the modify	modifier command.	
	command.	1) Check that the save method	pass
		doesn't apply the history. 2)	
		Check the output against the in-	
		put. 3) Check the history of the	
		output against the input.	
0.0		Test that the save method uses	pass
08	Control the method with a plist.	the filename which is stored in a	
		plist.	
		1) Check the output	pass
00	Test the save method with	Test the save method with differ-	pass
09	different complex PARFRAC	ent complex PARFRAC objects	
	objects	1) Check the output	pass

Table 237: Unit tests for parfrac/save.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



parfrac/setIunits			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists	pass
	Tests that the setJunits method	Test that the setJunits method	pass
02	works with a vector of PARFRAC objects as input.	works for a vector of PARFRAC objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'pfv' 2) Check that each output contains the correct data.	pass
03	Tests that the setIunits method works with a matrix of PARFRAC objects as input.	Test that the setIunits method works for a matrix of PARFRAC objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'pfm' 2) Check that each output contains the correct data.	pass
04	Tests that the setIunits method works with a list of PARFRAC objects as input.	Test that the setIunits method works for a list of PARFRAC ob- jects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the setIunits method works with a mix of different shaped PARFRAC objects as input.	Test that the setIunits method works with an input of matri- ces and vectors and single PAR- FRAC objects.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the setIunits method properly applies history and that the option 'internal'	Test that the result of applying the setIunits method can be pro- cessed back to an m-file.	pass
	suppresses the history.	 Check that the last entry in the history of 'out1' corresponds to 'setIunits'. 2) Check that the last entry in the history of 'out2' NOT corresponds to 'setIunits'. Check that the method re- build produces the same object as 'out'. 	pass
07	Tests that the setIunits methodcan modify the inputPARFRAC object.	Test that the setlunits method can modify the input PARFRAC object by calling with no output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

parfrac/setIunits			
		1) Check that 'pf3' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct iunits field	
00	Tests that the setIunits method	Test that the setIunits method	pass
00	can set the property with a	can modify the property 'iunits'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect iunits field 2) Check that	
		the method rebuild produces the	
		same object as 'out'.	
00	Check that the setIunits method	Call the method with a list of	pass
09	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 238: Unit tests for parfrac/setIunits.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



parfrac/setName			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists	pass
		have the correct parameters.	
02	Tests that the setName method	Test that the setName method	pass
	works with a vector of	works for a vector of PARFRAC	
	PARFRAC objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in out is the same as in	
		prv 2) Check that each output	
	Togta that the getName method	Test that the setName method	page
03	works with a matrix of	works for a matrix of DARERAC	pass
	PARERAC objects as input	objects as input	
	TAITING Objects as input.	1) Check that the number of el-	nase
		ements in 'out' is the same as in	pass
		'pfm' 2) Check that each output	
		contains the correct data.	
	Tests that the setName method	Test that the setName method	pass
04	works with a list of PARFRAC	works for a list of PARFRAC ob-	P
	objects as input.	jects as input.	
	0 1	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		contains the correct data.	
05	Tests that the setName method	Test that the setName method	pass
	works with a mix of different	works with an input of matri-	
	shaped PARFRAC objects as	ces and vectors and single PAR-	
	input.	FRAC objects.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		contains the correct data.	
06	lests that the setName method	lest that the result of applying	pass
	that the option 'internal'	accord back to an m file	
	suppresses the history	1) Check that the last entry in	nase
	suppresses the instory.	the history of 'out1' corresponds	pass
		to 'setName' 2) Check that the	
		last entry in the history of 'out?'	
		NOT corresponds to 'setName'.	
		3) Check that the method re-	
		build produces the same object	
		as 'out'.	
	Tests that the setName method	Test that the setName method	pass
07	can modify the input	can modify the input PARFRAC	
	PARFRAC object.	object by calling with no output.	


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

parfrac/setName			
		1) Check that 'pf3' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct name field	
08	Tests that the setName method	Test that the setName method	pass
00	can set the property with a	can modify the property 'name'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect name field 2) Check that	
		the method rebuild produces the	
		same object as 'out'.	
00	Check that the setName method	Call the method with a list of	pass
09	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 239: Unit tests for parfrac/setName.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



parfrac/setOunits			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the setOunits method	Test that the setOunits method	pass
	works with a vector of	works for a vector of PARFRAC	
	PARFRAC objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'pfv' 2) Check that each output	
		contains the correct data.	
03	Tests that the setOunits method	Test that the setOunits method	pass
	works with a matrix of	works for a matrix of PARFRAC	
	PARFRAC objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		pfm ²) Check that each output	
		contains the correct data.	
04	Tests that the setOunits method	Test that the setOunits method	pass
	works with a list of PARFRAC	works for a list of PARFRAC ob-	
	objects as input.	Jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		approximation the compact data	
	Trata that the actOurita method	The correct data.	magg
05	works with a mix of different	works with an input of matri	pass
	shaped PAPERAC objects as	works with an input of math-	
	input	ERAC objects	
	mput.	1) Check that the number of al	neg
		aments in 'out' is the same as in	pass
		input 2) Check that each output	
		contains the correct data	
	Tests that the setOunits method	Test that the result of apply-	nass
06	properly applies history and	ing the setOunits method can be	pass
	that the option 'internal'	processed back to an m-file	
	suppresses the history.	1) Check that the last entry in	pass
		the history of 'out1' corresponds	Paulo
		to 'setOunits'. 2) Check that the	
		last entry in the history of 'out2'	
		NOT corresponds to 'setOunits'.	
		3) Check that the method re-	
		build produces the same object	
		as 'out'.	
	Tests that the setOunits method	Test that the setOunits method	pass
07	can modify the input	can modify the input PARFRAC	
	PARFRAC object.	object by calling with no output.	
	· ·		



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

parfrac/setOunits			
		1) Check that 'pf3' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct ounits field	
08	Tests that the setOunits method	Test that the setOunits method	pass
00	can set the property with a	can modify the property 'ounits'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect ounits field 2) Check that	
		the method rebuild produces the	
		same object as 'out'.	
00	Check that the setOunits	Call the method with a list of	pass
09	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 240: Unit tests for parfrac/setOunits.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

parfrac/string			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the string method works with a vector of PARFRAC objects as input.	Test that the string method works for a vector of PARFRAC objects as input.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
03	Tests that the string method works with a matrix of PARFRAC objects as input.	Test that the string method works for a matrix of PARFRAC objects as input.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
04	Tests that the string method works with a list of PARFRAC objects as input.	Test that the string method works for a list of PARFRAC ob- jects as input.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
05	Tests that the string method works with a mix of different shaped PARFRAC objects as input.	Test that the string method works with an input of matri- ces and vectors and single PAR- FRAC objects.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
06	Tests that the string method properly applies history.	The method string doesn't change the data, thus it is not possible to check the history. Nothing to do.	pass
07	Tests that the string method doesn't work if the PARFRAC object have more than one history step	The method string throws an er- ror because the input object have more than one history step.	pass
1	moory buch.		pass

Table 241: Unit tests for parfrac/string.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

parfrac/type			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.0	Tests that the type method	Test that the type method works	pass
02	works with a vector of	for a vector of PARFRAC ob-	
	PARFRAC objects as input.	jects as input.	
		1) Check the rebuilt output.	pass
0.2	Tests that the type method	Test that the type method works	pass
03	works with a matrix of	for a matrix of PARFRAC ob-	
	PARFRAC objects as input.	jects as input.	
		1) Check the rebuilt output.	pass
0.4	Tests that the type method	Test that the type method works	pass
04	works with a list of PARFRAC	for a list of PARFRAC objects as	
	objects as input.	input.	
		1) Check the rebuilt output.	pass
05	Tests that the type method	Test that the type method works	pass
00	works with a mix of different	with an input of matrices and	
	shaped PARFRAC objects as	vectors and single PARFRAC	
	input.	objects.	
		1) Check the rebuilt output.	pass
06	Tests that the type method	The method type doesn't change	pass
00	properly applies history.	the data, thus it is not possible	
		to check the history. Nothing to	
		do.	
			pass

Table 242: Unit tests for parfrac/type.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pest/copy			
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	

Table 243: Unit tests for pest/copy.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pest/loadobj			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check the shape of the loaded	pass
		objects.	

Table 244: Unit tests for pest/loadobj.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pest/pest			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the [pest/pest] method works with a vector of objects as input.	Test that the [pest/pest] method works for a vector of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'vec' 2) Check that each output object contains the correct data.	pass
03	Tests that the [pest/pest] method works with a matrix of objects as input.	Test that the [pest/pest] method works for a matrix of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
04	Tests that the [pest/pest] method works with a list of objects as input.	Test that the [pest/pest] method works for a list of objects as in- put.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
05	Tests that the [pest/pest] method works with a mix of different arrays of objects as input.	Tests that the [pest/pest] method works with a mix of different arrays of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
06	Tests that the [pest/pest] method properly applies history.	Test that the result of applying the [pest/pest] method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[pest/pest]'. 2) Check that the re-built object is the same object as the input.	pass
60	Tests that the constructor method doesn't apply history to the read MAT-file constructor.	Tests that the constructor method doesn't apply history to the read MAT-file constructor.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pest/pest			
	Tests that the constructor	1) Check that the history is the same as the history of the saved object. Because save and load shouldn't add a history step. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
61	properly applies history to the read XML-file constructor.	erly applies history to the read XML-file constructor.	pass
		1) Check that the history is the same as the history of the saved object. Because save and load shouldn't add a history step. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
62	Tests that the constructor properly applies history in the struct constructor.	Tests that the constructor prop- erly applies history in the struct constructor.	pass
		1) Check that the last entry in the history of 'out' corresponds to the class name. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
64	Tests that the contructor properly applies history to the plist(filename) constructor.	Tests that the contructor prop- erly applies history to the plist(filename) constructor.	pass
		1) Check that the save method doesn't change the input ob- ject 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
65	Tests that the contructed object can be submitted and retrieved.	Tests that the contructed object can be submitted and retrieved.	pass
		the history of 'out' corresponds to the class name. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
68	Tests that the constructor properly applies history to the conn+Id constructor.	Tests that the constructor properly applies history to the conn+Id constructor.	pass
		1) Check that the last entry in the history of 'out' corresponds to class name. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass

Table 245: Unit tests for pest/pest.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pest/setChain			
minfo	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this a general setter method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
genericAnyShape			pass
Somerieringsmape			pass
genericAnvShape			pass
			pass
genericAnyShapeInte	rnal		pass
			pass
genericAnvShape			pass
			pass
genericAnyShape			pass
	1		pass
genericAnyShapeInte	ernal		pass
	Trata that the fract (act (heir)	Trate that the freet (ast () as in]	pass
genericList	Tests that the [pest/setChain]	lests that the [pest/setChain]	pass
	method works for a list of	method works for a list of objects	
	objects as input.	as input.	pagg
		amonts in 'out' is the same as in	pass
		'objes' 2) Check that each output	
		object contains the correct data	
	Tests that the [nest/setChain]	Tests that the [pest/setChain]	nass
genericList	method works for a list of	method works for a list of objects	pass
	objects as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	P
		'objs' 2) Check that each output	
		object contains the correct data.	
·	Tests that the [pest/setChain]	Tests that the [pest/setChain]	pass
genericList	method works for a list of	method works for a list of objects	
	objects as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'objs' 2) Check that each output	
		object contains the correct data.	
genericHistory	Tests that the [pest/setChain]	Test that the result of applying	pass
Scherioringtory	method properly applies history.	the [pest/setChain] method can	
		be processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[pest/setCham]'. 2) Check	
		that the re-built object is the	
		same object as the input.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pest/setChain			
mon onio Mo dife	Tests that the [pest/setChain]	Test that the [pest/setChain]	pass
genericmouny	method can modify the input	method can modify the input	
	AO.	object by calling with no out-	
		put and that the method doesn't	
		change the input of the function	
		notation (with a equal sign).	
		1) Check that the modified in-	pass
		put is changed by the method	
		2) Check that 'out' and 'obj_eq'	
		are now different. 3) Check that	
		'obj_eq' is not changed 4) Check	
		that out and amodi are the same	
mananiaOutnut	Check that the [pest/setChain]	Call the method with a list of	pass
genericOutput	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 246: Unit tests for pest/setChain.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pest/setChi2			
minfo	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this a general setter method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
genericAnyShape			pass
8			pass
genericAnvShape			pass
	1		pass
genericAnyShapeInte	rnal		pass
			pass
genericAnyShape			pass
			pass
genericAnyShape			pass
	1		pass
genericAnyShapeInte	rnal		pass
			pass
genericList	Tests that the [pest/setChi2]	Tests that the [pest/setChi2]	pass
	method works for a list of	method works for a list of objects	
	objects as input.	as input.	
		1) Check that the number of el-	pass
		ements in out is the same as in	
		object contains the correct data	
	Tosts that the [post/setChi2]	Tosts that the [post/setChi2]	nagg
genericList	method works for a list of	mothod works for a list of objects	pass
	objects as input	as input	
		1) Check that the number of el-	nass
		ements in 'out' is the same as in	pass
		'objs' 2) Check that each output	
		object contains the correct data.	
	Tests that the [pest/setChi2]	Tests that the [pest/setChi2]	pass
genericList	method works for a list of	method works for a list of objects	P
	objects as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	-
		'objs' 2) Check that each output	
		object contains the correct data.	
rononiaUistana	Tests that the [pest/setChi2]	Test that the result of applying	pass
genericfistory	method properly applies history.	the [pest/setChi2] method can	
		be processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[pest/setChi2]'. 2) Check that	
		the re-built object is the same	
		object as the input.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pest/setChi2			
rononiaMadify	Tests that the [pest/setChi2]	Test that the [pest/setChi2]	pass
genericimouny	method can modify the input	method can modify the input	
	AO.	object by calling with no out-	
		put and that the method doesn't	
		change the input of the function	
		notation (with a equal sign).	
		1) Check that the modified in-	pass
		put is changed by the method	
		2) Check that 'out' and 'obj_eq'	
		are now different. 3) Check that	
		'obj_eq' is not changed 4) Check	
		that out and amodi are the same	
ronomicOutput	Check that the [pest/setChi2]	Call the method with a list of	pass
genericOutput	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 247: Unit tests for pest/setChi2.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pest/setCorr			
minfo	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this a general setter method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
generic AnyShape			pass
generierinybhape			pass
genericAnyShape			pass
8			pass
genericAnyShapeInte	rnal		pass
			pass
genericAnvShape			pass
Of the June of the			pass
genericAnyShape			pass
	1		pass
genericAnyShapeInte	rnal		pass
			pass
genericList	Tests that the [pest/setCorr]	Tests that the [pest/setCorr]	pass
	method works for a list of	method works for a list of objects	
	objects as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		object contains the correct data	
	Tests that the [nest/setCom]	Tests that the [post/getCorrel	Dogg
genericList	method works for a list of	method works for a list of objects	pass
	objects as input	as input	
		1) Check that the number of el-	nass
		ements in 'out' is the same as in	pass
		'objs' 2) Check that each output	
		object contains the correct data	
	Tests that the [pest/setCorr]	Tests that the [pest/setCorr]	pass
genericList	method works for a list of	method works for a list of objects	Pass
	objects as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	1
		'objs' 2) Check that each output	
		object contains the correct data.	
	Tests that the [pest/setCorr]	Test that the result of applying	pass
genericHistory	method properly applies history.	the [pest/setCorr] method can be	-
		processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[pest/setCorr]'. 2) Check that	
		the re-built object is the same	
		object as the input.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pest/setCorr			
roponiaMadify	Tests that the [pest/setCorr]	Test that the [pest/setCorr]	pass
genericimouny	method can modify the input	method can modify the input	
	AO.	object by calling with no out-	
		put and that the method doesn't	
		change the input of the function	
		notation (with a equal sign).	
		1) Check that the modified in-	pass
		put is changed by the method	
		2) Check that 'out' and 'obj_eq'	
		are now different. 3) Check that	
		'obj_eq' is not changed 4) Check	
		that out and amodi are the same	
man ani a Outmut	Check that the [pest/setCorr]	Call the method with a list of	pass
genericOutput	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 248: Unit tests for pest/setCorr.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pest/setCov			
minfo	Tests that the getInfo call works for this a general setter method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
genericAnyShape			pass pass
genericAnyShape			pass pass
genericAnyShapeInt	ernal		pass pass
genericAnyShape			pass pass
genericAnyShape			pass
genericAnyShapeInt	ernal		pass pass
genericList	Tests that the [pest/setCov] method works for a list of objects as input.	Tests that the [pest/setCov] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericList	Tests that the [pest/setCov] method works for a list of objects as input.	Tests that the [pest/setCov] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericList	Tests that the [pest/setCov] method works for a list of objects as input.	Tests that the [pest/setCov] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericHistory	Tests that the [pest/setCov] method properly applies history.	Test that the result of applying the [pest/setCov] method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[pest/setCov]'. 2) Check that the re-built object is the same object as the input.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pest/setCov			
mon onio Mo dife	Tests that the [pest/setCov]	Test that the [pest/setCov]	pass
genericitiodity	method can modify the input	method can modify the input	
	AO.	object by calling with no output	
		and that the method doesn't	
		change the input of the function	
		notation (with a equal sign).	
		1) Check that the modified in-	pass
		put is changed by the method	
		2) Check that 'out' and 'obj_eq'	
		are now different. 3) Check that	
		'obj_eq' is not changed 4) Check	
		that out and amodi are the same	
rononicOutput	Check that the [pest/setCov]	Call the method with a list of	pass
genericOutput	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 249: Unit tests for pest/setCov.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pest/setDof			
minfo	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this a general setter method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
generic AnyShape			pass
generierinybhape			pass
genericAnyShape			pass
8			pass
genericAnyShapeInte	rnal		pass
			pass
genericAnyShape			pass
			pass
genericAnyShape			pass
· A C1 T (1		pass
genericAnyShapeInte	anal		pass
	Trata that the [read /astDaf]	Trate that the rest (ast Daf)	pass
genericList	I lests that the [pest/setDoi]	lests that the [pest/setDof]	pass
	shipets as input	inethod works for a list of objects	
	objects as input.	as input.	magg
		1) Check that the number of el-	pass
		'abig' 2) Check that each output	
		object contains the correct data	
	Tosts that the [nost/setDef]	Tosts that the [post/setDof]	nass
genericList	method works for a list of	method works for a list of objects	pass
	objects as input	as input	
		1) Check that the number of el-	nass
		ements in 'out' is the same as in	Pass
		'objs' 2) Check that each output	
		object contains the correct data.	
	Tests that the [pest/setDof]	Tests that the [pest/setDof]	pass
genericList	method works for a list of	method works for a list of objects	T
	objects as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'objs' 2) Check that each output	
		object contains the correct data.	
rononiaUistana	Tests that the [pest/setDof]	Test that the result of applying	pass
genericnistory	method properly applies history.	the [pest/setDof] method can be	
		processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[pest/setDof]'. 2) Check that	
		the re-built object is the same	
		object as the input.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pest/setDof			
mon onio Mo dife	Tests that the [pest/setDof]	Test that the [pest/setDof]	pass
genericimouny	method can modify the input	method can modify the input	
	AO.	object by calling with no output	
		and that the method doesn't	
		change the input of the function	
		notation (with a equal sign).	
		1) Check that the modified in-	pass
		put is changed by the method	
		2) Check that 'out' and 'obj_eq'	
		are now different. 3) Check that	
		'obj_eq' is not changed 4) Check	
		that out and amodi are the same	
mananiaOutnut	Check that the [pest/setDof]	Call the method with a list of	pass
genericOutput	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 250: Unit tests for pest/setDof.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pest/setDy			
minfo	Tests that the getInfo call works for this a general setter method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
genericAnyShape			pass pass
genericAnyShape			pass pass
genericAnyShapeInt	ernal		pass pass
genericAnyShape			pass pass
genericAnyShape			pass pass
genericAnyShapeInte	ernal		pass pass
genericList	Tests that the [pest/setDy] method works for a list of objects as input.	Tests that the [pest/setDy] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericList	Tests that the [pest/setDy] method works for a list of objects as input.	Tests that the [pest/setDy] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericList	Tests that the [pest/setDy] method works for a list of objects as input.	Tests that the [pest/setDy] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericHistory	Tests that the [pest/setDy] method properly applies history.	Test that the result of applying the [pest/setDy] method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[pest/setDy]'. 2) Check that the re-built object is the same object as the input.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pest/setDy			
mon onio Mo dife	Tests that the [pest/setDy]	Test that the [pest/setDy]	pass
genericitiodity	method can modify the input	method can modify the input	
	AO.	object by calling with no output	
		and that the method doesn't	
		change the input of the function	
		notation (with a equal sign).	
		1) Check that the modified in-	pass
		put is changed by the method	
		2) Check that 'out' and 'obj_eq'	
		are now different. 3) Check that	
		'obj_eq' is not changed 4) Check	
		that out and amodi are the same	
mon oni o Outmut	Check that the [pest/setDy]	Call the method with a list of	pass
genericOutput	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 251: Unit tests for pest/setDy.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pest/setModels			
minfo	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this a general setter method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
genericAnyShape			pass
Somerieringsmape			pass
genericAnvShape			pass
			pass
genericAnyShapeInte	rnal		pass
			pass
genericAnvShape			pass
			pass
genericAnyShape			pass
	1		pass
genericAnyShapeInte	ernal		pass
			pass
genericList	lests that the [pest/setModels]	lests that the [pest/setModels]	pass
	method works for a list of	method works for a list of objects	
	objects as input.	as input.	pagg
		amonts in 'out' is the same as in	pass
		'objes' 2) Check that each output	
		object contains the correct data	
	Tests that the [nest/setModels]	Tests that the [nest/setModels]	nass
genericList	method works for a list of	method works for a list of objects	pass
	objects as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	P
		'objs' 2) Check that each output	
		object contains the correct data.	
·	Tests that the [pest/setModels]	Tests that the [pest/setModels]	pass
genericList	method works for a list of	method works for a list of objects	
	objects as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'objs' 2) Check that each output	
		object contains the correct data.	
genericHistory	Tests that the [pest/setModels]	Test that the result of applying	pass
Scherioring	method properly applies history.	the [pest/setModels] method can	
		be processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to [pest/setModels]'. 2) Check	
		that the re-built object is the	
		same object as the input.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pest/setModels			
mon onio Mo dife	Tests that the [pest/setModels]	Test that the [pest/setModels]	pass
genericitiodity	method can modify the input	method can modify the input	
	AO.	object by calling with no out-	
		put and that the method doesn't	
		change the input of the function	
		notation (with a equal sign).	
		1) Check that the modified in-	pass
		put is changed by the method	
		2) Check that 'out' and 'obj_eq'	
		are now different. 3) Check that	
		'obj_eq' is not changed 4) Check	
		that out and amodi are the same	
rononicOutput	Check that the [pest/setModels]	Call the method with a list of	pass
genericOutput	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 252: Unit tests for pest/setModels.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pest/setNames			
minfo	Tests that the getInfo call works for this a general setter method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
genericAnyShape			pass pass
genericAnyShape			pass
genericAnyShape			pass
monomic A perChanaInt			pass
genericAnyShapeint	erna		pass
			pass
genericAnyShape			pass
			pass
genericAnyShape			pass
genericAnvShapeInt	ernal		pass
Seriering Strap error			pass
genericList	Tests that the [pest/setNames] method works for a list of objects as input.	Tests that the [pest/setNames] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericList	Tests that the [pest/setNames] method works for a list of objects as input.	Tests that the [pest/setNames] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericList	Tests that the [pest/setNames] method works for a list of objects as input.	Tests that the [pest/setNames] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericHistory	Tests that the [pest/setNames] method properly applies history.	Test that the result of applying the [pest/setNames] method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[pest/setNames]'. 2) Check that the re-built object is the same object as the input.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pest/setNames			
roponiaMadify	Tests that the [pest/setNames]	Test that the [pest/setNames]	pass
genericmodify	method can modify the input	method can modify the input	
	AO.	object by calling with no out-	
		put and that the method doesn't	
		change the input of the function	
		notation (with a equal sign).	
		1) Check that the modified in-	pass
		put is changed by the method	
		2) Check that 'out' and 'obj_eq'	
		are now different. 3) Check that	
		'obj_eq' is not changed 4) Check	
		that out and amodi are the same	
	Check that the [pest/setNames]	Call the method with a list of	pass
genericOutput	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 253: Unit tests for pest/setNames.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pest/setPdf			
minfo	Tests that the getInfo call works for this a general setter method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
genericAnyShape			pass pass
genericAnyShape			pass pass
genericAnyShapeInt	ernal		pass pass
genericAnyShape			pass pass
genericAnyShape			pass
genericAnyShapeInt	ernal		pass pass
genericList	Tests that the [pest/setPdf] method works for a list of objects as input.	Tests that the [pest/setPdf] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericList	Tests that the [pest/setPdf] method works for a list of objects as input.	Tests that the [pest/setPdf] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericList	Tests that the [pest/setPdf] method works for a list of objects as input.	Tests that the [pest/setPdf] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericHistory	Tests that the [pest/setPdf] method properly applies history.	Test that the result of applying the [pest/setPdf] method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[pest/setPdf]'. 2) Check that the re-built object is the same object as the input.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pest/setPdf			
rononiaMadify	Tests that the [pest/setPdf]	Test that the [pest/setPdf]	pass
genericimouny	method can modify the input	method can modify the input	
	AO.	object by calling with no output	
		and that the method doesn't	
		change the input of the function	
		notation (with a equal sign).	
		1) Check that the modified in-	pass
		put is changed by the method	
		2) Check that 'out' and 'obj_eq'	
		are now different. 3) Check that	
		'obj_eq' is not changed 4) Check	
		that out and amodi are the same	
mananiaOutnut	Check that the [pest/setPdf]	Call the method with a list of	pass
genericOutput	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 254: Unit tests for pest/setPdf.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pest/setY			
minfo	Tests that the getInfo call works for this a general setter method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
genericAnyShape			pass pass
genericAnyShape			pass pass
genericAnyShapeInt	ernal		pass pass
genericAnyShape			pass pass
genericAnyShape			pass pass
genericAnyShapeInt	ernal		pass pass
genericList	Tests that the [pest/setY] method works for a list of objects as input.	Tests that the [pest/setY] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericList	Tests that the [pest/setY] method works for a list of objects as input.	Tests that the [pest/setY] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericList	Tests that the [pest/setY] method works for a list of objects as input.	Tests that the [pest/setY] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericHistory	Tests that the [pest/setY] method properly applies history.	Test that the result of applying the [pest/setY] method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[pest/setY]'. 2) Check that the re-built object is the same object as the input.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pest/setY			
rononiaMadify	Tests that the [pest/setY]	Test that the [pest/setY] method	pass
genericimouny	method can modify the input	can modify the input object by	
	AO.	calling with no output and that	
		the method doesn't change the	
		input of the function notation	
		(with a equal sign).	
		1) Check that the modified in-	pass
		put is changed by the method	
		2) Check that 'out' and 'obj_eq'	
		are now different. 3) Check that	
		'obj_eq' is not changed 4) Check	
		that out and amodi are the same	
mon oni o Outmut	Check that the [pest/setY]	Call the method with a list of	pass
genericOutput	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 255: Unit tests for pest/setY.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pest/setYunits			
minfo	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this a general setter method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
generic Any Shape			pass
genericrinyshape			pass
generic AnyShape			pass
Senerierinysnape			pass
generic AnyShape			pass
genericrinyshape			pass
generic AnyShape			pass
genericrinyshape			pass
genericAnyShapeInt	ernal		pass
			pass
genericAnyShape			pass
Sonorioring snapo			pass
genericAnyShape			pass
Semerating Smape			pass
genericAnyShapeInt	ernal		pass
			pass
genericList	Tests that the [pest/setYunits]	Tests that the [pest/setYunits]	pass
Souchering	method works for a list of	method works for a list of objects	
	objects as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'objs' 2) Check that each output	
		object contains the correct data.	
genericList	Tests that the [pest/setYunits]	Tests that the [pest/setYunits]	pass
0	method works for a list of	method works for a list of objects	
	objects as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		objs' 2) Check that each output	
	Trata that the [rt/tV't]	object contains the correct data.	
genericList	lests that the [pest/set Yunits]	lests that the [pest/setYunits]	pass
	method works for a list of	method works for a list of objects	
	objects as input.	as input.	Decc
		amonta in 'out' is the same as in	pass
		'chia' 2) Choole that each autout	
		object contains the correct data	
	Trata that the [rt /tV-r.'t]	Test that the provide of even is	nc
genericHistory	rests that the [pest/set runits]	the [next /act Vurite] method	pass
· ·	method property applies history.	the [pest/set runits] method can	
		be processed back.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pest/setYunits			
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[pest/setYunits]'. 2) Check	
		that the re-built object is the	
		same object as the input.	
mon oni o Mo difu	Tests that the [pest/setYunits]	Test that the [pest/setYunits]	pass
genericmodify	method can modify the input	method can modify the input	
	AO.	object by calling with no out-	
		put and that the method doesn't	
		change the input of the function	
		notation (with a equal sign).	
		1) Check that the modified in-	pass
		put is changed by the method	-
		2) Check that 'out' and 'obj_eq'	
		are now different. 3) Check that	
		'obj_eq' is not changed 4) Check	
		that out and amodi are the same	
	Check that the [pest/setYunits]	Call the method with a list of	pass
genericOutput	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	1
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 256:	Unit	tests	for	$pest_{/}$	/setY	units.
------------	------	------------------------	----------------------	------------	-------	--------



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



plist/append			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the append method works with a vector of PLIST objects as input.	Test that the append method appends all input PLIST objects to one object.	pass
		1) Check that the output is one PLIST object. 2) Check that the output PLIST contains all key/value pairs.	pass
03	Tests that the append method works with a matrix of PLIST objects as input.	Test that the append method appends all input PLIST objects to one object.	pass
		1) Check that the output is one PLIST object. 2) Check that the output PLIST contains all key/value pairs.	pass
04	Tests that the append method works with a list of PLIST objects as input.	Test that the append method appends all input PLIST objects to one object.	pass
		1) Check that the output is one PLIST object. 2) Check that the output PLIST contains all key/value pairs.	pass
05	Tests that the append method works with a mix of different shaped PLIST objects as input.	Test that the append method appends all input PLIST objects to one object.	pass
		1) Check that the output is one PLIST object. 2) Check that the output PLIST contains all key/value pairs.	pass
06	Tests that the append method applies the modify command	Test that the append method can modify the input PLIST by call- ing with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass
		1) Check that 'pl1' and 'plin' are now different. 2) Check that 'ain' append the key/value pair.	pass
07	Test the append method with different input variants for the key/value pair.	Test that the append method accepts plists-objects, param- objects or direct key/value pairs as an input.	pass
		1) Check that the output have all key/value pairs.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

plist/append			
08	Test that the append method	Test that the append method ap-	pass
00	appends the key always in	pends the key always in capital	
	capital letter	letter	
		1) Check that the key is in capi-	pass
		tal letters	
00	Test the append method in a	The append method throwns an	pass
09	negative case.	error if a user tries to append a	
		key which already exist.	
		1) Check that the negative case	pass
		doesn't change the input object.	

Table 257: Unit tests for plist/append.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

plist/char			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.0	Tests that the char method	Test that the char method works	pass
02	works with a vector of PLIST	for a vector of PLIST objects as	
	objects as input.	input.	
		1) Check that the output contain	pass
		at least each object name	
02	Tests that the char method	Test that the char method works	pass
05	works with a matrix of PLIST	for a matrix of PLIST objects as	
	objects as input.	input.	
		1) Check that the output contain	pass
		at least each object name	
04	Tests that the char method	Test that the char method works	pass
04	works with a list of PLIST	for a list of PLIST objects as in-	
	objects as input.	put.	
		1) Check that the output contain	pass
		at least each object name	
05	Tests that the char method	Test that the char method works	pass
00	works with a mix of different	with an input of matrices and	
	shaped PLIST objects as input.	vectors and single PLIST ob-	
		jects.	
		1) Check that the output contain	pass
		at least each object name	

Table 258: Unit tests for plist/char.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



plist/combine			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the combine method works with a vector of PLIST objects as input.	Test that the combine method combines all input PLIST ob- jects to one object.	pass
		1) Check that the output is one PLIST object. 2) Check that the output PLIST contains all key/value pairs.	pass
03	Tests that the combine method works with a matrix of PLIST objects as input.	Test that the combine method combines all input PLIST ob- jects to one object.	pass
		1) Check that the output is one PLIST object. 2) Check that the output PLIST contains all key/value pairs.	pass
04	Tests that the combine method works with a list of PLIST objects as input.	Test that the combine method combines all input PLIST ob- jects to one object.	pass
		1) Check that the output is one PLIST object. 2) Check that the output PLIST contains all key/value pairs.	pass
05	Tests that the combine method works with a mix of different shaped PLIST objects as input.	Test that the combine method combines all input PLIST ob- jects to one object.	pass
		1) Check that the output is one PLIST object. 2) Check that the output PLIST contains all key/value pairs.	pass
06	Tests that the combine method applies the modify command	Test that the combine method can modify the input PLIST by calling with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass
		1) Check that 'pl1' and 'plin' are now different. 2) Check that 'ain' combine the key/value pair.	pass
07	Test the combine method doen't overwrite existing keys.	Duplicate parameters which are given priority in the order in which they appear in the input.	pass
		1) Uneck that the output have all key/value pairs in the order they appear.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



plist/comb	oine
------------	------

Table 259: Unit tests for plist/combine.


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

plist/copy			
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	

Table 260: Unit tests for plist/copy.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

plist/display			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the display method	Test that the display method	pass
02	works with a vector of PLIST	works for a vector of PLIST ob-	
	objects as input.	jects as input.	
		1) Check that the output contain	pass
		at least each object name	
03	Tests that the display method	Test that the display method	pass
05	works with a matrix of PLIST	works for a matrix of PLIST ob-	
	objects as input.	jects as input.	
		1) Check that the output contain	pass
		at least each object name	
04	Tests that the display method	Test that the display method	pass
04	works with a list of PLIST	works for a list of PLIST objects	
	objects as input.	as input.	
		1) Check that the output contain	pass
		at least each object name	
05	Tests that the display method	Test that the display method	pass
00	works with a mix of different	works with an input of matrices	
	shaped PLIST objects as input.	and vectors and single PLIST ob-	
		jects as.	
		1) Check that the output contain	pass
		at least each object name	

Table 261: Unit tests for plist/display.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

plist/eq			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the eq method works	Test that the eq method works	pass
02	with a vector of PLIST objects	for a vector of PLIST objects as	
	as input.	input. Test the positive and the	
		negative case.	
		1) Check the output of the eq	pass
		function.	
03	Tests that the eq method works	Test that the eq method works	pass
	with a matrix of PLIST objects	for a matrix of PLIST objects as	
	as input.	input. Test the positive and the	
		negative case.	
		1) Check the output of the eq	pass
		function.	
04	Tests that the eq method works	The eq method doesn't works for	pass
01	with a list of PLIST objects as	a list of PLIST objects as input.	
	input.	Nothing to do.	
			pass
05	Tests that the eq method works	The eq method doesn't works for	pass
	with a mix of different shaped	a list of PLIST objects as input.	
	PLIST objects as input.	Nothing to do.	
			pass
06	Test the eq method with an	Test the eq method with the ex-	pass
	exception list. With the LTPDA	ception 'name'. It is necessary to	
	toolbox 2.0 it is only possible to	add 'created' to the exception list	
	test the exception list with	because pl is created at an other	
	properties where a public set	time.	
	method exist.	1) Check the output.	pass
07	Test the eq method with an	Test that the eq method uses the	pass
	exception list which is in a plist.	exception list in a plist.	
		1) Check the output.	pass

Table 262: Unit tests for plist/eq.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

plist/find			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the find method	Test that it is possible to use	pass
02	applies the modify command.	modify command and show that	
		this method ignoring case.	
		1) Check that out13 have all the	pass
		same object.	
02	Test that the find method	Test that the find method take	pass
00	accepts a plist.	the 'key' from a 'search' plist.	
		1) Check that the output have	pass
		the correct value.	
0.4	Test the find method returns an	Test the find method returns an	pass
04	empty array if the 'key' doesn't	empty array if the 'key' doesn't	
	exist in the PLIST.	exist in the PLIST.	
		1) Check that the output is an	pass
		empty array.	

Table 263: Unit tests for plist/find.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

· · · · · · · · · · · · · · · · · · ·			
plist/get			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests the get method of the	Test that the get returns returns	pass
02	plist class.	the value of the specified prop-	
		erty. Do this for all properties of	
		the PLIST object.	
		1) Check the correct value of the	pass
		output	
0.2	Tests that the get method works	Test that the get returns returns	pass
00	with a plist.	the value of the specified prop-	
		erty which is defined in a plist.	
		1) Check the correct value of the	pass
		output	
04	Tests the get method of the	Test that the get throws an error	pass
04	plist class.	if the input are more than one	
		PLIST object.	
		1) Nothing to test	pass

Table 264: Unit tests for plist/get.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

plist/isparam			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the isparam method	Tests that the isparam method	pass
02	works with a vector of PLIST	works with a vector of PLIST ob-	
	objects as input.	jects as input.	
		1) Check that the output have	pass
		the same size as the input. 2)	
		Check the output.	
0.2	Tests that the isparam method	Tests that the isparam method	pass
03	works with a matrix of PLIST	works with a matrix of PLIST	
	objects as input.	objects as input.	
		1) Check that the output have	pass
		the same size as the input. 2)	
		Check the output.	
0.4	Tests that the isparam method	Tests that the isparam method	pass
04	works with a list of PLIST	works with a list of PLIST ob-	
	objects as input.	jects as input.	
		1) Check that the output have	pass
		the same size as the input. 2)	
		Check the output.	
05	Tests that the isparam method	Tests that the isparam method	pass
00	works with a mix of different	works with a mix of different	
	shaped PLIST objects as input.	shaped PLIST objects as input.	
		1) Check that the output have	pass
		the same size as the input. 2)	
		Check the output.	
06	Tests that the isparam method	Test that the isparam method	pass
00	applies the modify command	can used in the modifier form.	
		1) Test that plin doesn't change.	pass
07	Test the isparam method with a	Test the isparam method with	pass
01	positiv match.	different input.	
		1) Check the output	pass

Table 265: Unit tests for plist/isparam.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

plist/isprop			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the isprop method works with a vector of PLIST objects as input.	Test that the isprop method works for a vector of PLIST ob- jects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'plv' 2) Check that each output contains the correct data.	pass
03	Tests that the isprop method works with a matrix of PLIST objects as input.	Test that the isprop method works for a matrix of PLIST ob- jects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'plm' 2) Check that each output contains the correct data.	pass
04	Tests that the isprop method works with a list of PLIST objects as input.	Test that the isprop method works for a list of PLIST objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the isprop method works with a mix of different shaped PLIST objects as input.	Test that the isprop method works with an input of matrices and vectors and single PLIST ob- jects.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the isprop method works for each property.	Test that the isprop method works for the properties: 'params', 'created', 'creator', 'name'	pass
		tains the correct data.	Pass
07	Test the negative case and the not function command.	Test that the isprop method retrun false for a unknown prop- erty and for methods of the ob- ject.	pass
		1) Check that each output con- tains the correct data.	pass

Table 266: Unit tests for plist/isprop.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



plist/loadobj			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check the shape of the loaded	pass
		objects.	

Table 267: Unit tests for plist/loadobj.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

plist/ne			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the ne method works with a vector of PLIST objects as input.	Test that the ne method works for a vector of PLIST objects as input. Test the positive and the negative case.	pass
		1) Check the output of the ne function.	pass
03	Tests that the ne method works with a matrix of PLIST objects as input.	Test that the ne method works for a matrix of PLIST objects as input. Test the positive and the negative case.	pass
		1) Check the output of the ne function.	pass
04	Tests that the ne method works with a list of PLIST objects as input.	The ne method doesn't works for a list of PLIST objects as input. Nothing to do.	pass
05	Tests that the ne method works with a mix of different shaped PLIST objects as input.	The ne method doesn't works for a list of PLIST objects as input. Nothing to do.	pass
06	Tests that the ne method properly applies history.	The ne method doesn't change the PLIST object, thus will no history added. Nothing to do	pass pass
07	Test the ne method with an exception list. The function plist/ne use the function plist/eq so it is not necessary to check all possibilities of the exception list.	Test the ne method with the exception 'name'. It is necessary to add 'created' to the exception list because pl is created at an other time. 1) Check that each output contains the correct data	pass
08	Test the ne method with an exception list which is in a plist.	Test that the ne method uses the exception list in a plist.	pass
		1) Check that each output con- tains the correct data.	pass

Table 268: Unit tests for plist/ne.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

plist/nparams			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the nparams method	Test that the nparams method	pass
02	works with a vector of PLIST	returns the number of PARAM	
	objects as input.	objects in the PLIST objects.	
		1) Check that the number of out-	pass
		puts is the same as the number	
		of input PLIST objects. 2) Check	
		the output.	
0.2	Tests that the nparams method	Test that the nparams method	pass
05	works with a vector of PLIST	returns the number of PARAM	
	objects as input.	objects in the PLIST objects.	
		1) Check that the number of out-	pass
		puts is the same as the number	
		of input PLIST objects. 2) Check	
		the output.	
04	Tests that the nparams method	Test that the nparams method	pass
04	works with a list of PLIST	returns the number of PARAM	
	objects as input.	objects in the PLIST objects.	
		1) Check that the number of out-	pass
		puts is the same as the number	
		of input PLIST objects. 2) Check	
		the output.	
05	Tests that the nparams method	Test that the nparams method	pass
05	works with a mix of different	returns the number of PARAM	
	shaped PLIST objects as input.	objects in the PLIST objects.	
		1) Check that the number of out-	pass
		puts is the same as the number	
		of input PLIST objects. 2) Check	
		the output.	
06	Tests that the nparams method	Test that the nparams method	pass
	applies the modify command	can used in the modifier style.	
		1) Check the output.	pass

Table 269: Unit tests for plist/nparams.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

plist/parse			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		1) Check that matters call re	
		1) Check that gettino can re-	pass
		cases 2) Check that all plists	
		have the correct parameters	
	Tests that the parse method	The parse method is an internal	nase
02	works with a vector of PLIST	method and it doesn't work for	pass
	objects as input	vector of PLIST objects as an in-	
		nut	
		1) Nothing to test	nass
	Tests that the parse method	The parse method is an internal	pass
03	works with a matrix of PLIST	method and it doesn't work for	pass
	objects as input.	matrix of PLIST objects as an	
	osjeets as mpatt	input.	
		1) Nothing to test	pass
	Tests that the parse method	The parse method is an internal	pass
04	works with a list of PLIST	method and it doesn't work for	
	objects as input.	list of PLIST objects as an input.	
		1) Nothing to check.	pass
05	Tests that the parse method	The parse method is an internal	pass
00	works with a mix of different	method and it doesn't work for	
	shaped PLIST objects as input.	different shaped PLIST objects	
		as input.	
		1) Nothing to test.	pass
06	Tests that the parse method	Test that the parse method can	pass
	with non-dependent elements.	convert non-dependent elements.	
		1) Check the right number of pa-	pass
		rameter in out. 2) Check that the	
		correct values are converted.	
07	Tests that the parse method	Test that the parse method can	pass
	with dependent elements.	convert dependent elements.	
		1) Check the right number of pa-	pass
		rameter in out. 2) Check that the	
	Togta that upper /lower coss	Testa that upper/lever	
08	lests that upper/lower case	lests that upper/lower case	pass
	method	mothed	
	method.	1) Check the right number of pa	neg
		ramator in out 2) Check that the	pass
		correct values are converted	
1		content values are converted.	1

Table 270: Unit tests for plist/parse.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

plist/plist			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists	pass
		The table to the correct parameters.	
02	works with a vector of PLIST	with a vector of PLIST objects	pass
	objects as input	as input	
	objects as input.	1) Check that the shape of the	nase
		output PLISTs is the same as the	pass
		input shape. 2) Check that each	
		output PLIST is a copy of the	
		input PLIST.	
03	Tests that the plist method	Test that the plist method works	pass
05	works with a matrix of PLIST	with a matrix of PLIST objects	
	objects as input.	as input.	
		1) Check that the shape of the	pass
		output PLISTs is the same as the	
		input shape. 2) Check that each	
		output PLIST is a copy of the	
	Trata that the ulist mathed	The the the relief with a drawning	
04	works with a list of DI IST	with a list of DLIST abjects as	pass
	works with a list of PLIST	input	
	objects as input.	1) Check that the number of ele-	nase
		ments in 'out' is the same of the	pass
		number in the input. 2) Check	
		that each output PLIST is a copy	
		of the input PLIST.	
05	Tests that the plist method	Test that the plist method works	pass
05	works with a mix of different	with a mix of different shaped	
	shaped PLISTs as input.	PLISTs as input.	
		1) Check that the number of ele-	pass
		ments in 'out' is the same of the	
		number in the input. 2) Check	
		of the input PLIST is a copy	
		Tosts the MAT file constructor	negg
06	Tests the MAT-file constructor.	1) Check that the saved object is	pass
		the same as the loaded object is	Pass
		Tests the XML-file constructor	pass
07	Tests the XML-file constructor.	1) Check that the saved object is	pass
		the same as the loaded object.	
0.0	Tests that the plist method for	Tests that the plist method for a	pass
08	a struct as an input.	struct as an input.	
		2) Check the built object.	pass
00	Tosts the param constructor	Tests the param constructor.	pass
. 09	rests the param constructor.	·	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

plist/plist			
		1) Check that the output con-	pass
		tains the input parameter ob-	
		jects	
10	Tests the plist(filename)	Tests the plist(filename) con-	pass
10	constructor.	structor.	
		1) Check that the loaded object	pass
		is the same as the saved object	
11	Tests the plist(conn)	Tests the plist(conn) construc-	pass
	constructor.	tor.	
		1) Check the retrieved object	pass
		against the submitted object.	
10	Tests the conn + Id constructor	Tests the conn+Id constructor.	pass
12		1) Check the retrieved object	pass
		against the submitted object.	
19	Tests the key /velue constructor	Tests the key/value constructor	pass
10	Tests the key/value constructor	1) Check the right number of	pass
		parameter in the PLIST object	
		2) Check the parameter in the	
		PLIST object 3) Check that the	
		key is in capital letter.	

Table 271: Unit tests for plist/plist.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

plist/plist2cmds			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.0	Tests that the plist2cmds	The plist2cmds method doesn't	pass
02	method works with a vector of	works for a vector of PLIST ob-	
	PLIST objects as input.	jects as input.	
		1) Nothing to test.	pass
0.2	Tests that the plist2cmds	The plist2cmds method doesn't	pass
05	method works with a matrix of	works for a matrix of PLIST ob-	
	PLIST objects as input.	jects as input.	
		1) Nothing to test.	pass
04	Tests that the plist2cmds	The plist2cmds method doesn't	pass
04	method works with a list of	works for a list of PLIST objects	
	PLIST objects as input.	as input.	
		1) Nothing to test.	pass
05	Tests that the plist2cmds	The plist2cmds method doesn't	pass
00	method works with a mix of	works with an input of matrices	
	different shaped PLIST objects	and vectors and single PLIST ob-	
	as input.	jects.	
		1) Nothing to test	pass
06	Tests that the plist2cmds	Create a plist with all possible	pass
00	method accepts different objects	objects for the 'val' property.	
	for the 'val' property.	1) Check that the output is a ex-	pass
		ecutable plist2cmds	

Table 272: Unit tests for plist/plist2cmds.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

plist/pset			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the pset method	Test that the pset method set	pass
02	works with a vector of PLIST	or add a key/value pair to all	
	objects as input.	PLIST objects in the vector.	
		1) Check that the shape of the	pass
		output is the same as the shape	
		of the input 2) Check that the	
		output PLIST contains the new	
		key/value pair.	
0.2	Tests that the pset method	Test that the pset method set	pass
03	works with a matrix of PLIST	or add a key/value pair to all	-
	objects as input.	PLIST objects in the matrix.	
		1) Check that the shape of the	pass
		output is the same as the shape	1
		of the input 2) Check that the	
		output PLIST contains the new	
		kev/value pair.	
2.4	Tests that the pset method	Test that the pset method set	pass
04	works with a list of PLIST	or add a key/value pair to all	1
	objects as input.	PLIST objects in the input.	
	<i>J I</i>	1) Check that the shape of the	pass
		output is the same as the shape	I
		of the input 2) Check that the	
		output PLIST contains the new	
		key/value pair.	
	Tests that the pset method	Test that the pset method set	pass
05	works with a mix of different	or add a key/value pair to all	P
	shaped PLIST objects as input.	PLIST objects in the input.	
		1) Check that the shape of the	pass
		output is the same as the shape	Pass
		of the input 2) Check that the	
		output PLIST contains the new	
		key/value pair	
	Tests that the nset method	Test that the nset method can	nass
06	applies the modify command	modify the input PLIST by call-	Pass
	applies the mouly command	ing with no output and that the	
		mg with no output and that the	
		of the function notation (with a	
		or the function notation (with a	
		equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

plist/pset			
		1) Check that 'out' and 'pleq'	pass
		are now different. 2) Check that	
		'plmo' and 'out' are the same.	
		3) Check that 'out' and 'plmo'	
		have the new key/value pair 4)	
		Check that pleq don't have the	
		new key/value pair	
07	Test the pset method works for	Test that the pset method ac-	pass
07	different input variants for the	cepts param-objects or direct	
	key/value pair.	key/value pairs as an input.	
		1) Check that the output have all	pass
		key/value pairs.	
08	Test the pset method in the	Create an example where the	pass
00	setting and appending case.	pset method set a key to a new	
		value and an example where pset	
		append the new key/value pair	
		1) Check the number of param-	pass
		etes in the output. 2) Check the	
		new key/value pair	
00	Test that the pset method sets	Test that the pset method sets	pass
	the key always in capital letter	the key always in capital letter	
		1) Check that the key is in capi-	pass
		tal letters	

Table 273: Unit tests for plist/pset.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

plist/remove			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the remove method	Test that the remove method re-	pass
02	works with a vector of PLIST	move the 'key' from all PLIST	
	objects as input.	objects in the vector.	
		1) Check that the shape of the	pass
		output is the same as the shape	
		of the input 2) Check that the	
		output PLIST doesn't contains	
		the key.	
03	Tests that the remove method	Test that the remove method re-	pass
00	works with a matrix of PLIST	move the 'key' from all PLIST	
	objects as input.	objects in the matrix.	
		1) Check that the shape of the	pass
		output is the same as the shape	
		of the input 2) Check that the	
		output PLIST doesn't contains	
		the key.	
04	Tests that the remove method	Test that the remove method re-	pass
04	works with a list of PLIST	move the 'key' from all PLIST	
	objects as input.	objects of the input.	
		1) Check that the shape of the	pass
		output is the same as the shape	
		of the input 2) Check that the	
		output PLIST doesn't contains	
		the key.	
05	Tests that the remove method	Test that the remove method re-	pass
00	works with a mix of different	move the 'key' from all PLIST	
	shaped PLIST objects as input.	objects of the input.	
		1) Check that the shape of the	pass
		output is the same as the shape	
		of the input 2) Check that the	
		output PLIST doesn't contains	
		the key.	
06	Tests that the remove method	Test that the remove method can	pass
	applies the modify command	modify the input PLIST by call-	
		ing with no output and that the	
		method doesn't change the input	
		of the function notation (with a	
		equal sign).	
	-		



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

plist/remove			
		1) Check that 'out' and 'pleq'	pass
		are now different. 2) Check that	
		'plmo' and 'out' are the same. 3)	
		Check that 'out' and 'plmo' don't	
		have the key 4) Check that pleq	
		doesn't have the key	
07	Test the remove method that it	Test the remove method that it	pass
07	removes the parameter which is	removes the parameter which is	
	defined as an index or as a key	defined as an index or as a key	
		1) Check the output.	pass
00	Test the remove method in a	The remove method doesn't	pass
08	negative case that the key is not	throwns an error if the key	
	in the paramter list.	doesn't exist in the parameter	
		list.	
		1) Nothing to test.	pass

Table 274: Unit tests for plist/remove.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

plist/save			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the save method works with a vector of PLIST objects as input.	Test that the save method works for a vector of PLIST objects as input. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out1' and 'out2' are the same as in 'plv' 2) Check that the loaded objects are the same as the saved objects. 3) The out- puts 'out1' and 'out2' must be the same.	pass
03	Tests that the save method works with a matrix of PLIST objects as input.	Test that the save method works for a matrix of PLIST objects as input. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of ele- ments in 'out1' and 'out2' are the same as in 'plm' 2) Check that the loaded objects are the same as the saved objects. 3) The out- puts 'out1' and 'out2' must be the same.	pass
04	Tests that the save method works with a list of PLIST objects as input.	Test that the save method works for a list of PLIST objects as in- put. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out1' and 'out2' are the same as in the list 2) Check that the loaded objects are the same as the saved objects. 3) The outputs 'out1' and 'out2' must be the same.	pass
05	Tests that the save method works with a mix of different shaped PLIST objects as input.	Test that the save method works with an input of matrices and vectors and single PLIST ob- jects. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output PLIST object contains the cor- rect data.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

plist/save			
06	Tests that the save method	Use the save method with the	pass
00	works with the modify	modifier command.	
	command.	1) Check that the save method	pass
		doesn't change the input PLIST	
		object. 2) Check the output	
		against the input except.	
07	Control the method with a plict	Test that the save method uses	pass
07	Control the method with a plist.	the filename which is stored in a	
		plist.	
		1) Check the output	pass

Table 275: Unit tests for plist/save.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

plist/setName			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minto object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the setName method	Test that the setName method	pass
	works with a vector of PLIST	works for a vector of PLIST ob-	
	objects as input.	Jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		plv 2) Check that each output	
	Trate that the set Name mathed	The correct data.	
03	Tests that the setName method	rest that the setName method	pass
	works with a matrix of PLIST	works for a matrix of PLIST OD-	
	objects as input.	1) Check that the number of al	
		1) Check that the humber of el-	pass
		'nlm' 2) Check that each output	
		contains the correct data	
	Tests that the setName method	Tost that the setName method	page
04	works with a list of PLIST	works for a list of PLIST objects	pass
	objects as input	as input	
	objects as input.	1) Chock that the number of al	nase
		aments in 'out' is the same as in	pass
		input 2) Check that each output	
		contains the correct data	
	Tests that the setName method	Test that the setName method	nase
05	works with a mix of different	works with an input of matrices	pass
	shaped PLIST objects as input	and vectors and single PLIST ob-	
	shaped i his i objects as input.	iects	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	pass
		input. 2) Check that each output	
		contains the correct data.	
	Tests that the setName method	Test that the setName method	pass
06	can modify the input PLIST	can modify the input PLIST ob-	
	object.	ject by calling with no output.	
		1) Check that 'pl2' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct name field	
07	Tests that the setName method	Test that the setName method	pass
07	can set the property with a	can modify the property 'name'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect name field	
0.9	Check that the setName method	Call the method with a list of	pass
08	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable.	
	single variable.	<u> </u>	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



plist/setName		
	1) Check that the output con-	pass
	tains the right humber of objects	

Table 276: Unit tests for plist/setName.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

plist/string			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the string method	Test that the string method	pass
02	works with a vector of PLIST	works for a vector of PLIST ob-	
	objects as input.	jects as input.	
		1) Check that the output is a ex-	pass
		ecutable string. 2) Check the cor-	
		rect number of rout 3) Check the	
		rebuild objects.	
03	Tests that the string method	Test that the string method	pass
00	works with a matrix of PLIST	works for a matrix of PLIST ob-	
	objects as input.	jects as input.	
		1) Check that the output is a ex-	pass
		ecutable string. 2) Check the cor-	
		rect number of rout 3) Check the	
		rebuild objects.	
0.4	Tests that the string method	Test that the string method	pass
04	works with a list of PLIST	works for a list of PLIST objects	
	objects as input.	as input.	
		1) Check that the output is a ex-	pass
		ecutable string. 2) Check the cor-	
		rect number of rout 3) Check the	
		rebuild objects.	
05	Tests that the string method	Test that the string method	pass
00	works with a mix of different	works with an input of matrices	
	shaped PLIST objects as input.	and vectors and single PLIST ob-	
		jects.	
		1) Check that the output is a ex-	pass
		ecutable string. 2) Check the cor-	
		rect number of rout 3) Check the	
		rebuild objects.	
06	Tests that the string method	The method string throws an er-	pass
	doesn't work if the PLIST	ror because the input object have	
	object have more than one	more than one history step.	
	history step.		pass
07	Tests that the string method	Create a plist with all possible	pass
	accepts different objects for the	objects for the 'val' property.	
	'val' property.	1) Check that the output is a ex-	pass
		ecutable string	

Table 277: Unit tests for plist/string.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/char			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the char method works with a vector of PZMODEL objects as input.	Test that the char method works for a vector of PZMODEL ob- jects as input.	pass
		1) Check that the output contain at least each object name	pass
03	Tests that the char method works with a matrix of PZMODEL objects as input.	Test that the char method works for a matrix of PZMODEL ob- jects as input.	pass
		1) Check that the output contain at least each object name	pass
04	Tests that the char method works with a list of PZMODEL objects as input.	Test that the char method works for a list of PZMODEL objects as input.	pass
		1) Check that the output contain at least each object name	pass
05	Tests that the char method works with a mix of different shaped PZMODEL objects as input.	Test that the char method works with an input of matrices and vectors and single PZMODEL objects.	pass
		1) Check that the output contain at least each object name	pass
06	Tests that the char method properly applies history.	The method char doesn't change the data, thus it is not possible to check the history. Nothing to do.	pass
			pass

Table 278: Unit tests for pzmodel/char.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/copy			
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	

Table 279: Unit tests for pzmodel/copy.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/created			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the created method works with a vector of PZMODEL objects as input.	Test that the created method works for a vector of PZMODEL objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'pzv' 2) Check that each output contains the correct data.	pass
03	Tests that the created method works with a matrix of PZMODEL objects as input.	Test that the created method works for a matrix of PZMODEL objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'pzm' 2) Check that each output contains the correct data.	pass
04	Tests that the created method works with a list of PZMODEL objects as input.	Test that the created method works for a list of PZMODEL ob- jects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the created method works with a mix of different shaped PZMODEL objects as input.	Test that the created method works with an input of matri- ces and vectors and single PZ- MODEL objects.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the created method properly applies history	This method doesn't change the input object, thus no history is added to the object.	pass
07	Tests that the created method can be used with the modify command.	 1) Nothing to check. Tests that the created method can be used with the modify command. 1) Check the single object 2) 	pass pass pass
08	Tests that the created method retruns always a well defined time object even for an empty input object.	Test that the created method with an empty 'PZMODEL ob- ject	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/created		
	1) Check that the output is a time object with a ell defined time.	pass

Table 280: Unit tests for pzmodel/created.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/creator			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.0	Tests that the creator method	Test that the creator method	pass
02	works with a vector of	works for a vector of PZMODEL	
	PZMODEL objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	-
		'pzv' 2) Check that each output	
		contains the correct data.	
	Tests that the creator method	Test that the creator method	pass
03	works with a matrix of	works for a matrix of PZMODEL	1
	PZMODEL objects as input.	objects as input.	
	J 1	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'pzm' 2) Check that each output	
		contains the correct data.	
	Tests that the creator method	The creator method doesn't work	pass
04	works with a list of PZMODEL	for a list of PZMODEL objects as	pass
	objects as input.	input.	
		1) Nothing to test.	pass
	Tests that the creator method	The creator method doesn't work	pass
05	works with a mix of different	with different shaped input ob-	pass
	shaped PZMODEL objects as	iects.	
	input.	1) Nothing to test	pass
	Tests that the creator method	This method doesn't change the	pass
06	properly applies history	input object thus no history is	pass
		added to the object	
		1) Nothing to check	nass
	Tests that the creator method	Tests that the creator method	pass
07	can be used with the modify	can be used with the modify	pass
	command	command	
		1) Check the single object 2	nass
		Check the matrix object	pass
	Tests that the creator method	Test that the creator method	nass
08	retrung all	uses the option 'all' direct or in	Pass
	creator(s)/modifier(s) which are	a plist. The test file must have	
	in the history	the modifier 'first' 'second' and	
		'third'	
		1) Check that out1 contains only	
		and croater 2) Check that out?	pass
		one creator 2) Check that Out2	
		contain more creator/modifier	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/creator			
09	Tests the negative case for the option 'all'.	Test that the creator method throws an error if the option 'all' is used in connection with a matrix/vector of PZMODEL ob- jects.	pass
		1) Nothing to test.	pass

Table 281: Unit tests for pzmodel/creator.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/display			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the display method	Test that the display method	pass
02	works with a vector of	works for a vector of PZMODEL	
	PZMODEL objects as input.	objects as input.	
		1) Check that the output contain	pass
		at least each object name	
03	Tests that the display method	Test that the display method	pass
00	works with a matrix of	works for a matrix of PZMODEL	
	PZMODEL objects as input.	objects as input.	
		1) Check that the output contain	pass
		at least each object name	
04	Tests that the display method	Test that the display method	pass
L L L L L L L L L L L L L L L L L L L	works with a list of PZMODEL	works for a list of PZMODEL ob-	
	objects as input.	jects as input.	
		1) Check that the output contain	pass
		at least each object name	
05	Tests that the display method	Test that the display method	pass
	works with a mix of different	works with an input of matri-	
	shaped PZMODEL objects as	ces and vectors and single PZ-	
	input.	MODEL objects as.	
		1) Check that the output contain	pass
		at least each object name	
06	Tests that the display method	The method display doesn't	pass
	properly applies history.	change the data, thus it is not	
		possible to check the history.	
		Nothing to do.	
			pass

Table 282: Unit tests for pzmodel/display.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/eq			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the eq method works with a vector of PZMODEL objects as input.	Test that the eq method works for a vector of PZMODEL ob- jects as input. Test the positive and the negative case.	pass
	Tests that the eq method works	function.	pass
03	with a matrix of PZMODEL objects as input.	for a matrix of PZMODEL objects as input. Test the positive and the negative case.	pass
		1) Check the output of the eq function.	pass
04	Tests that the eq method works with a list of PZMODEL objects as input.	The eq method doesn't works for a list of PZMODEL objects as in- put. Nothing to do.	pass
			pass
05	Tests that the eq method works with a mix of different shaped PZMODEL objects as input.	The eq method doesn't works for a list of PZMODEL objects as in- put. Nothing to do.	pass
			pass
06	Tests that the eq method properly applies history.	The eq method doesn't change the PZMODEL object, thus will no history added. Nothing to do	pass
			pass
07	Test the eq method with an exception list. With the LTPDA toolbox 2.0 it is only possible to test the exception list with properties where a public set method exist.	Test the eq method with the exception 'name'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because pzm is created at an other time. 1) Check the output.	pass
00	Test the eq method with an	Test the eq method with the ex-	pass
08	exception list. With the LTPDA toolbox 2.0 it is only possible to test the exception list with properties where a public set method exist.	ception 'iunits'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because pzm is created at an other time. 1) Check the output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/eq			
00	Test the eq method with an	Test the eq method with the ex-	pass
03	exception list. With the LTPDA	ception 'ounits'. Use the option	
	toolbox 2.0 it is only possible to	'internal' to suppress the history.	
	test the exception list with	It is necessary to add 'created' to	
	properties where a public set	the exception list because pzm is	
	method exist.	created at an other time.	
		1) Check the output.	pass
10	Test the eq method with an	Test that the eq method uses the	pass
	exception list which is in a plist.	exception list in a plist.	
		1) Check the output.	pass

Table 283: Unit tests for pzmodel/eq.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/get			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests the get method of the	Test that the get returns returns	pass
02	pzmodel class.	the value of the specified prop-	
		erty. Do this for all properties of	
		the PZMODEL object.	
		1) Check the correct value of the	pass
		output	
0.2	Tests that the get method works	Test that the get returns returns	pass
05	with a plist.	the value of the specified prop-	
		erty which is defined in a plist.	
		1) Check the correct value of the	pass
		output	
04	Tests the get method of the	Test that the get throws an error	pass
	pzmodel class.	if the input are more than one	
		PZMODEL object.	
		1) Nothing to test	pass

Table 284: Unit tests for pzmodel/get.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/getlower	Freq		
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests the getlowerFreq method	Test that the getlowerFreq re-	pass
02	of the pzmodel class.	turns the lowest frequence of the	
		lowest pole or zero in the model.	
		1) Check the output	pass
03	Tests the getlowerFreq method	Test that the getlowerFreq	pass
	of the pzmodel class.	throws an error if the input are	
		more than one pzmodel.	
		1) Nothing to test	pass

Table 285: Unit tests for pzmodel/getlowerFreq.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/getupper	Freq		
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.9	Tests the getupperFreq method	Test that the getupperFreq re-	pass
02	of the pzmodel class.	turns the lowest frequence of the	
		lowest pole or zero in the model.	
		1) Check the output	pass
03	Tests the getupperFreq method	Test that the getupperFreq	pass
	of the pzmodel class.	throws an error if the input are	
		more than one pzmodel.	
		1) Nothing to test	pass

Table 286: Unit tests for pzmodel/getupperFreq.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/index			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the index method works with a vector of PZMODEL objects as input.	Test that the index method works for a vector of PZMODEL objects as input. The following indexing should work: $I = [1 \ 2 \ 3]$ or $(I/J) = [(1,1), (1,2), (1,3)]$	pass
	Tests that the index method	1) Check that the index method selects the correct object.	pass
03	works with a matrix of PZMODEL objects as input.	works for a matrix of PZMODEL objects as input. The following indexing should work: $I = [1 \ 3 \ 5]$ or $(I/J) = [(1,1), (1,2), (1,3)] [$ $2 \ 4 \ 6] [(2,1), (2,2), (2,3)]$	
		1) Check that the index method selects the correct object.	pass
04	Tests that the index method works with a list of PZMODEL objects as input.	The index method doesn't work for a list of PZMODEL objects as input.	pass
05	Tests that the index method properly applies history.	 Test that the result of index have an additional history step. 1) Check that the last entry in the history of 'out' corresponds to 'index'. 	pass pass
06	Tests that the index method works for the modifier command.	Tests that the index method works for the modifier command.1) Check that the history-plist contains the used indices. 2) Check that the index method selects the correct object	pass pass
07	Control the method with a plist.	Test that the index method can be controled with a plist. 1) Check that the history-plist contains the used indices. 2) Check that the index method se- lects the correct object	pass
08	Test that the index method selects more objects if I have more indices.	Test that the index method se- lects more objects if I have more indices.	pass
		1) Check that the history-plist contains the used indices. 2) Check that the index method se- lects the correct object	pass


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/	/index
----------	--------

Table 287: Unit tests for pzmodel/index.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/isprop			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the isprop method works with a vector of PZMODEL objects as input.	Test that the isprop method works for a vector of PZMODEL objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'pzv' 2) Check that each output contains the correct data.	pass
03	Tests that the isprop method works with a matrix of PZMODEL objects as input.	Test that the isprop method works for a matrix of PZMODEL objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'pzm' 2) Check that each output contains the correct data.	pass
04	Tests that the isprop method works with a list of PZMODEL objects as input.	Test that the isprop method works for a list of PZMODEL ob- jects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the isprop method works with a mix of different shaped PZMODEL objects as input.	Test that the isprop method works with an input of matri- ces and vectors and single PZ- MODEL objects.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the isprop method properly applies history.	The method isprop doesn't change the object, thus it is not necessary to apply history.	pass
07	Tests that the isprop method works for each property.	Test that the isprop method works for the properties: 'gain', 'poles', 'zeros', 'iunits', 'ounits', 'hist', 'name'	pass
		1) Check that each output con- tains the correct data.	pass
08	Test the negative case and the not function command.	Test that the isprop method retrun false for a unknown prop- erty and for methods of the ob- ject.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/isprop		
	1) Check that each output con-	pass
	tains the correct data.	

Table 288: Unit tests for pzmodel/isprop.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/loadobj			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check the shape of the loaded	pass
		objects.	

Table 289: Unit tests for pzmodel/loadobj.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

pzmodel/mrdiv	vide		
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the mrdivide method works with a vector of PZMODELs as input.	Test that the mrdivide method works for a vector of PZMOD- ELs as input.	pass
		1) Check that the output is exact one PZMODEL object. 2) Check the gain of the output 3) Check the poles of the output 4) Check the zeros of the output	pass
03	Tests that the mrdivide method works with a matrix of PZMODELs as input.	Tests that the rdivie method works with a matrix of PZMOD- ELs as input.	pass
		1) Check that the output is exact one PZMODEL object. 2) Check the gain of the output 3) Check the poles of the output 4) Check the zeros of the output	pass
04	Tests that the mrdivide method works with a list of PZMODELs as input.	Tests that the mrdivide method works with a list of PZMODELs as input.	pass
		1) Check that the output is exact one PZMODEL object. 2) Check the gain of the output 3) Check the poles of the output 4) Check the zeros of the output	pass
05	Tests that the mrdivide method works with a mix of different shaped PZMODELs as input.	Tests that the mrdivide method works with a mix of different shaped PZMODELs as input.	pass
		1) Check that the output is exact one PZMODEL object. 2) Check the gain of the output 3) Check the poles of the output 4) Check the zeros of the output 5) Check the rebuilt object	pass
06	Tests that the mrdivide method properly applies history.	Test that the result of applying the mrdivide method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'mrdivide'. 2) Check that re- built object is the same object as the input.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/mrdivide			
07	Check that the mrdivide	Check that the mrdivide method	pass
07	method only divide PZMODELs	only divide PZMODELs with the	
	with the same output units.	same output units. Check also	
		the negative case.	
		1) Check the I-/O-units	pass

Table 290: Unit tests for pzmodel/mrdivide.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

pzmodel/mtimes			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the mtimes method works with a vector of PZMODELs as input.	Test that the mtimes method works for a vector of PZMOD- ELs as input.	pass
		1) Check that the output is exact one PZMODEL object. 2) Check the gain of the output 3) Check the poles of the output 4) Check the zeros of the output	pass
03	Tests that the mtimes method works with a matrix of PZMODELs as input.	Tests that the mtimes method works with a matrix of PZMOD- ELs as input.	pass
		1) Check that the output is exact one PZMODEL object. 2) Check the gain of the output 3) Check the poles of the output 4) Check the zeros of the output	pass
04	Tests that the mtimes method works with a list of PZMODELs as input.	Tests that the mtimes method works with a list of PZMODELs as input.	pass
		1) Check that the output is exact one PZMODEL object. 2) Check the gain of the output 3) Check the poles of the output 4) Check the zeros of the output	pass
05	Tests that the mtimes method works with a mix of different shaped PZMODELs as input.	Tests that the mtimes method works with a mix of different shaped PZMODELs as input.	pass
		1) Check that the output is exact one PZMODEL object. 2) Check the gain of the output 3) Check the poles of the output 4) Check the zeros of the output 5) Check the rebuilt object	pass
06	Tests that the mtimes method properly applies history.	Test that the result of applying the mtimes method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'times'. 2) Check that re-built object is the same object as the input.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/mtimes			
07	Check that the mtimes method	Check that the mtimes method	pass
07	only multiply PZMODELs with	only multiply PZMODELs with	
	the same output units.	the same output units. Check	
		also the negative case.	
		1) Check the I-/O-units	pass

Table 291: Unit tests for pzmodel/mtimes.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/ne			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the ne method works with a vector of PZMODEL objects as input.	Test that the ne method works for a vector of PZMODEL ob- jects as input. Test the positive and the negative case.	pass
		1) Check the output of the ne function.	pass
03	Tests that the ne method works with a matrix of PZMODEL objects as input.	Test that the ne method works for a matrix of PZMODEL ob- jects as input. Test the positive and the negative case.	pass
		1) Check the output of the ne function.	pass
04	Tests that the ne method works with a list of PZMODEL objects as input.	The ne method doesn't works for a list of PZMODEL objects as in- put. Nothing to do.	pass
			pass
05	Tests that the ne method works with a mix of different shaped PZMODEL objects as input.	The ne method doesn't works for a list of PZMODEL objects as in- put. Nothing to do.	pass
			pass
06	Tests that the ne method properly applies history.	The ne method doesn't change the PZMODEL object, thus will no history added. Nothing to do	pass
			pass
07	Test the ne method with an exception list. The function pzmodel/ne use the function pzmodel/eq so it is not necessary to check all possibilities of the exception list.	Test the ne method with the exception 'name'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because pzm is created at an other time.	pass
		1) Oneck that each output con-	pass
08	Test the ne method with an exception list which is in a plist.	Test that the ne method uses the exception list in a plist.	pass
		1) Check that each output con- tains the correct data.	pass

Table 292: Unit tests for pzmodel/ne.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/pzmodel			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the pzmodel method works with a vector of PZMODEL objects as input.	Test that the prodel method works with a vector of PZ- MODEL objects as input.	pass
		 Check that the shape of the output PZMODELs is the same as the input shape. Check that each output PZMODEL is a copy of the input PZMODEL. Check that the copy have an additional history step. 	pass
03	Tests that the pzmodel method works with a matrix of PZMODEL objects as input.	Test that the pzmodel method works with a matrix of PZ- MODEL objects as input.	pass
		 Check that the shape of the output PZMODELs is the same as the input shape. 2) Check that each output PZMODEL is a copy of the input PZMODEL. Check that the copy have an additional history step. 	pass
04	Tests that the pzmodel method works with a list of PZMODEL objects as input.	Test that the pzmodel method works with a list of PZMODEL objects as input.	pass
		 Check that the number of elements in 'out' is the same of the number in the input. 2) Check that each output PZMODEL is a copy of the input PZMODEL. Check that the copy have an additional history step. 	pass
05	Tests that the pzmodel method works with a mix of different shaped PZMODELs as input.	Test that the pzmodel method works with a mix of different shaped PZMODELs as input.	pass
		 Check that the number of elements in 'out' is the same of the number in the input. 2) Check that each output PZMODEL is a copy of the input PZMODEL. Check that the copy have an additional history step. 	pass
06	Tests that the pzmodel method properly applies history.	Test that the result of applying the pzmodel method can be pro- cessed back.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

pzmodel/pzmodel			
		1) Check that the last entry in the history of 'out' corre- sponds to 'pzmodel'. 2) Check that the method rebuild pro- duces the same object as 'out'.	pass
07	Tests that the pzmodel method properly applies history to the copy constructor.	Test that the output can be pro- cessed back with the 'rebuild' method. Test the constructor with a different number of in- puts.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'pzmodel'. 2) Check that the original objects are not changed by the setter function 3) Check that the method rebuild pro- duces the same object as 'out'.	pass
08	Tests that the pzmodel method properly applies history to the read MAT-file constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the loaded object is the same as the saved object. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
09	Tests that the pzmodel method properly applies history to the read XML-file constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the loaded object is the same as the saved object. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
10	Tests that the pzmodel method properly applies history to the read FIL-file constructor.	Read the FIL file which is cre- ated from LISO. Test that the output can be processed back with the 'rebuild' method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'pzmodel'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
11	Tests that the pzmodel method properly applies history to the struct constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'pzmodel'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/pzmodel			
12	Tests that the pzmodel from a constant properly applies history.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'pzmodel'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
13	Tests that the pzmodel method properly applies history to the rational constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'pzmodel'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
14	Tests that the pzmodel method properly applies history to the	Test that the output can be pro- cessed back to an m-file.	pass
	plist(filename) constructor.	1) Check that the save method doesn't change the input ob- ject 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
15	Tests that the PZMODEL method properly applies history to the plist(conn) constructor.	Test that the output can be processed back with the rebuild method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'pzmodel'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
16	Tests that the PZMODEL method properly applies history to the plist(rational) constructor	Test that the output can be processed back with the rebuild method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'pzmodel'. 2) Check the algo- rithm 3) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
17	Tests that the PZMODEL method properly applies history to the plist(gain poles zeros)	Test that the output can be processed back with the rebuild method.	pass
	constructor.	1) Check that the last entry in the history of 'out' corresponds to 'pzmodel'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

pzmodel/pzmodel			
18	Tests that the PZMODEL method properly applies history to the plist(<plist-object>)</plist-object>	Test that the output can be processed back with the rebuild method.	pass
	constructor.	1) Check that the last entry in the history of 'out' corresponds to 'pzmodel'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
19	Tests that the PZMODEL method properly applies history to the conn+Id constructor.	Test that the output can be processed back with the rebuild method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'pzmodel'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
20	Tests that the PZMODEL method properly applies history to the gain + poles + zeros	Test that the output can be processed back with the rebuild method.	pass
	constructor.	1) Check that the last entry in the history of 'out' corresponds to 'pzmodel'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
21	Tests that the PZMODEL method properly applies history to the gain + poles + zeros +	Test that the output can be processed back with the rebuild method.	pass
	name constructor.	1) Check that the last entry in the history of 'out' corresponds to 'pzmodel'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
22	Tests that the PZMODEL method properly applies history to the gain + poles + zeros +	Test that the output can be processed back with the rebuild method.	pass
	iunits + ounits constructor.	1) Check that the last entry in the history of 'out' corresponds to 'pzmodel'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass

Table 293: Unit tests for pzmodel/pzmodel.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

pzmodel/rdivide			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the rdivide method	Test that the rdivide method	pass
02	works with a vector of	works for a vector of PZMOD-	
	PZMODELs as input.	ELs as input.	
		1) Check that the output is exact	pass
		one PZMODEL object. 2) Check	
		the gain of the output 3) Check	
		the poles of the output 4) Check	
		the zeros of the output	
0.2	Tests that the rdivide method	Tests that the rdivie method	pass
05	works with a matrix of	works with a matrix of PZMOD-	
	PZMODELs as input.	ELs as input.	
		1) Check that the output is exact	pass
		one PZMODEL object. 2) Check	
		the gain of the output 3) Check	
		the poles of the output 4) Check	
		the zeros of the output	
0.4	Tests that the rdivide method	Tests that the rdivide method	pass
04	works with a list of PZMODELs	works with a list of PZMODELs	
	as input.	as input.	
	-	1) Check that the output is exact	pass
		one PZMODEL object. 2) Check	
		the gain of the output 3) Check	
		the poles of the output 4) Check	
		the zeros of the output	
05	Tests that the rdivide method	Tests that the rdivide method	pass
05	works with a mix of different	works with a mix of different	
	shaped PZMODELs as input.	shaped PZMODELs as input.	
		1) Check that the output is exact	pass
		one PZMODEL object. 2) Check	
		the gain of the output 3) Check	
		the poles of the output 4) Check	
		the zeros of the output 5) Check	
		the rebuilt object	
0.0	Tests that the rdivide method	Test that the result of applying	pass
06	properly applies history.	the rdivide method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	-
		to 'rdivide'. 2) Check that re-	
		built object is the same object as	
		the input.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/rdivide			
07	Check that the rdivide method	Check that the rdivide method	pass
07	only divide PZMODELs with	only divide PZMODELs with the	
	the same output units.	same output units. Check also	
		the negative case.	
		1) Check the I-/O-units	pass

Table 294: Unit tests for pzmodel/rdivide.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/rebuild			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the rebuild method	Test that the rebuild method	pass
02	works with a vector of	works for a vector of PZMODEL	
	PZMODEL objects as input.	objects as input.	
		1) Check the rebuilt output.	pass
03	Tests that the rebuild method	Test that the rebuild method	pass
00	works with a matrix of	works for a matrix of PZMODEL	
	PZMODEL objects as input.	objects as input.	
		1) Check the rebuilt output.	pass
04	Tests that the rebuild method	Test that the rebuild method	pass
01	works with a list of PZMODEL	works for a list of PZMODEL ob-	
	objects as input.	jects as input.	
		1) Check the rebuilt output.	pass
05	Tests that the rebuild method	Test that the rebuild method	pass
	works with a mix of different	works with an input of matri-	
	shaped PZMODEL objects as	ces and vectors and single PZ-	
	input.	MODEL objects.	
		1) Check the rebuilt output.	pass
06	Tests that the rebuild method	The method rebuild doesn't	pass
	properly applies history.	change the data, thus it is not	
		possible to check the history.	
		Nothing to do.	
			pass
07	Check that the rebuild method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 295: Unit tests for pzmodel/rebuild.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/resp			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set	pass
		1) Check that matinfa call no	magg
		1) Check that gettino can re-	pass
		cases 2) Check that all plists	
		have the correct parameters	
	Togta that the reap method	Tast that the rosp method works	page
02	works with a vector of	for a vector of P7MODEL ob	pass
	PZMODEL objects as input	jocts as input. Tost the method	
	I ZMODEL objects as input.	with an output and with no out-	
		put (a diagram must appear)	
		1) Test the right number of lines	page
		in the diagram 2) Check that the	pass
		number of elements in 'out' is the	
		same as in 'pgy' 3) Check that	
		same as in pzv 3) Check that	
		the correct data	
	Togta that the reap method	Tast that the resp method works	page
03	works with a matrix of	for a matrix of PZMODEL ob	pass
	PZMODEL objects as input	ior a matrix of FZMODEL 00-	
	PZMODEL objects as input.	jects as input. Test the method	
		with an output and with no out-	
		1) Test the right number of lines	maga
		in the diagram 2) Check that the	pass
		m the diagram. 2) Check that the	
		number of elements in out is the	
		same as in prinat 5) Check that	
		the compact date	
		The correct data.	
04	lests that the pzmat method	lest that the resp method works	pass
	works with a list of PZMODEL	for a list of PZMODEL objects	
	objects as input.	as input. Test the method with	
		diagram must appear)	
		1) Test the right number of lines	
		1) Test the right number of lines	pass
		in the diagram. 2) Check that the	
		number of elements in out is the	
		same as in 'pzmat' 3) Check that	
		each output PZMODEL contains	
		the correct data.	
05	1 ests that the resp method	Lest that the resp method works	pass
	works with a mix of different	with an input of matrices and	
	snaped PZMODEL objects as	vectors and single PZMODEL	
	input.	objects. Lest the method with an	
		output and with no output (a di-	
		agram must appear)	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

pzmodel/resp			
		1) Test the right number of lines in the diagram. 2) Check that the number of elements in 'out' is the same as in 'pzmat' 3) Check that each output PZMODEL contains the correct data.	pass
06	properly applies history.	Test that the result of apply- ing the resp method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'resp'. 2) Check that re-built object is the same object as the input.	pass
07	Tests that modify command plots the response into a diagram.	Tests that modify command plots the response into a dia- gram.	pass
08	Test the shape of the output.	Test that the output AO of the resp method keeps the shape of the used input f vector.	pass
	Check that the resp method	 Check that the shape of the data doesn't change. Call the method with a list of 	pass
09	pass back the output objects to a list of output variables or to a single variable.	output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
10	Check that the resp method uses the x-data of an input AO for f-vector.	Call the method with different method to pass an AO in. 1) Check the output 2) Check that the 'rebuild' method pro-	pass pass
	Check that the resp method	duces the same object as 'out'.	nass
11	uses the specified f-vector to compute the response.	method to pass an f-vector in. 1) Check the output 2) Check that the 'rebuild' method pro-	pass
12	Check that the resp method uses the specified f1, f2, and nf	duces the same object as 'out'. Call the method with different method to pass f1, f2, and nf in.	pass
	to compute the response.	1) Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'.	pass
13	Check that the resp method uses the specified f1, f2, and nf to compute the response.	Call the method with different method to pass f1, f2, and nf in.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/resp		
	1) Check the output 2) Check	pass
	that the 'rebuild' method pro-	
	duces the same object as 'out'.	

Table 296: Unit tests for pzmodel/resp.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/save			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the save method works with a vector of PZMODEL objects as input.	Test that the save method works for a vector of PZMODEL ob- jects as input. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of ele- ments in 'out1' and 'out2' are the same as in 'pzv' 2) Check that the loaded objects are the same as the saved objects. 3) The out- puts 'out1' and 'out2' must be the same.	pass
03	Tests that the save method works with a matrix of PZMODEL objects as input.	Test that the save method works for a matrix of PZMODEL ob- jects as input. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out1' and 'out2' are the same as in 'pzm' 2) Check that the loaded objects are the same as the saved objects. 3) The outputs 'out1' and 'out2' must be the same.	pass
04	Tests that the save method works with a list of PZMODEL objects as input.	Test that the save method works for a list of PZMODEL objects as input. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out1' and 'out2' are the same as in the list 2) Check that the loaded objects are the same as the saved objects. 3) The outputs 'out1' and 'out2' must be the same.	pass
05	Tests that the save method works with a mix of different shaped PZMODEL objects as input.	Test that the save method works with an input of matrices and vectors and single PZMODEL objects. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each out- put PZMODEL object contains the correct data.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/save			
06	Tests that the save method	Test that the result of apply-	pass
	properly applies history.	ing the save method can be pro-	
		cessed back to an m-file. Do this	
		for both extensions 'mat' and	
		'xml'	
		1) Check that the history applies	pass
		to the output object. Check that	
		save doesn't add a history step to	
		the input object. 2) Check that	
		the read object doesn't contain	
		the save + load history steps.	
		3) Check that the method re-	
		build produces the same object	
		as 'out'.	
07	Tests that the save method	Use the save method with the	pass
	works with the modify	modifier command.	
	command.	1) Check that the save method	pass
		doesn't apply the history. 2)	
		Check the output against the in-	
		put. 3) Check the history of the	
		output against the input.	
08	Control the method with a plist	Test that the save method uses	pass
08	Control the method with a plist.	the filename which is stored in a	
		plist.	
		1) Check the output	pass
00	Test the save method with	Save all common PZMODEL ob-	pass
03	common PZMODEL objects.	jects with both extensions.	
		1) Check the output	pass

Table 297: Unit tests for pzmodel/save.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/setDelay			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the setDelay method	Test that the setDelay method	pass
	works with a vector of	works for a vector of PZMODEL	
	PZMODEL objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		pzv ²) Check that each output	
	Trata that the act Dalars we the d	The correct data.	
03	Tests that the setDelay method	rest that the setDelay method	pass
	DZMODEL shipsts as input	works for a matrix of PZMODEL	
	PZMODEL objects as input.	1) Check that the number of al	D 0.00
		amonta in 'out' is the same as in	pass
		'ngm' 2) Check that each output	
		contains the correct data	
	Tests that the setDelay method	Tost that the setDelay method	nase
04	works with a list of PZMODEI	works for a list of PZMODEL ob	pass
	objects as input	iects as input	
		1) Check that the number of el-	nass
		ements in 'out' is the same as in	pass
		input 2) Check that each output	
		contains the correct data	
	Tests that the setDelay method	Test that the setDelay method	pass
05	works with a mix of different	works with an input of matri-	pass
	shaped PZMODEL objects as	ces and vectors and single PZ-	
	input.	MODEL objects.	
	1	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		contains the correct data.	
06	Tests that the setDelay method	Test that the result of applying	pass
00	properly applies history and	the setDelay method can be pro-	
	that the option 'internal'	cessed back to an m-file.	
	suppresses the history.	1) Check that the last entry in	pass
		the history of 'out1' corresponds	
		to 'setDelay'. 2) Check that the	
		last entry in the history of 'out2'	
		NOT corresponds to 'setDelay'.	
		3) Check that the method re-	
		build produces the same object	
		as 'out'.	
07	Tests that the setDelay method	Test that the setDelay method	pass
· · ·	can modify the input	can modify the input PZMODEL	
	PZMODEL object.	object by calling with no output.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/setDelay			
		1) Check that 'pz5' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct delay field	
08	Tests that the setDelay method	Test that the setDelay method	pass
00	can set the property with a	can modify the property 'delay'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect delay field 2) Check that	
		the method rebuild produces the	
		same object as 'out'.	
00	Check that the setDelay method	Call the method with a list of	pass
09	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 298: Unit tests for pzmodel/setDelay.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/setIunits	5		
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the setlunits method	Test that the setlunits method	pass
	works with a vector of	works for a vector of PZMODEL	
	PZMODEL objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		pzv ²) Check that each output	
	Trate that the action its mathed	The correct data.	
03	Tests that the settunits method	rest that the setiumts method	pass
	PZMODEL shisets of input	works for a matrix of PZMODEL	
	PZMODEL objects as input.	1) Check that the number of al	-
		amonta in 'out' is the same as in	pass
		'ngm' 2) Check that each output	
		contains the correct data	
	Tosts that the setlupits method	Tost that the softward method	nase
04	works with a list of PZMODEI	works for a list of PZMODEL ob	pass
	objects as input	iects as input	
	objects as input.	1) Check that the number of el-	nass
		ements in 'out' is the same as in	pass
		input 2) Check that each output	
		contains the correct data	
	Tests that the setIunits method	Test that the setlunits method	pass
05	works with a mix of different	works with an input of matri-	Pass
	shaped PZMODEL objects as	ces and vectors and single PZ-	
	input.	MODEL objects.	
	1	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		contains the correct data.	
06	Tests that the setIunits method	Test that the result of applying	pass
00	properly applies history and	the setIunits method can be pro-	
	that the option 'internal'	cessed back to an m-file.	
	suppresses the history.	1) Check that the last entry in	pass
		the history of 'out1' corresponds	
		to 'setIunits'. 2) Check that the	
		last entry in the history of 'out2'	
		NOT corresponds to 'setIunits'.	
		3) Check that the method re-	
		build produces the same object	
		as 'out'.	
07	Tests that the setIunits method	Test that the setIunits method	pass
	can modify the input	can modify the input PZMODEL	
	PZMODEL object.	object by calling with no output.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

pzmodel/setIunits			
		1) Check that 'pz5' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct iunits field	
08	Tests that the setIunits method	Test that the setIunits method	pass
00	can set the property with a	can modify the property 'iunits'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect iunits field 2) Check that	
		the method rebuild produces the	
		same object as 'out'.	
00	Check that the setIunits method	Call the method with a list of	pass
09	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 299: Unit tests for pzmodel/setIunits.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/setNam	le		
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters	pass
02	Tests that the setName methodworks with a vector ofPZMODEL objects as input.	Test that the setName method works for a vector of PZMODEL objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'pzv' 2) Check that each output contains the correct data.	pass
03	Tests that the setName methodworks with a matrix ofPZMODEL objects as input.	Test that the setName method works for a matrix of PZMODEL objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'pzm' 2) Check that each output contains the correct data.	pass
04	Tests that the setName method works with a list of PZMODEL objects as input.	Test that the setName method works for a list of PZMODEL ob- jects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the setName method works with a mix of different shaped PZMODEL objects as input.	Test that the setName method works with an input of matri- ces and vectors and single PZ- MODEL objects.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the setName method properly applies history and that the option 'internal'	Test that the result of applying the setName method can be pro- cessed back to an m-file.	pass
	suppresses the history.	 Check that the last entry in the history of 'out1' corresponds to 'setName'. 2) Check that the last entry in the history of 'out2' NOT corresponds to 'setName'. Check that the method re- build produces the same object as 'out'. 	pass
07	Tests that the setName methodcan modify the inputPZMODEL object.	Test that the setName method can modify the input PZMODEL object by calling with no output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

pzmodel/setName			
		1) Check that 'pz5' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct name field	
08	Tests that the setName method	Test that the setName method	pass
00	can set the property with a	can modify the property 'name'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect name field 2) Check that	
		the method rebuild produces the	
		same object as 'out'.	
00	Check that the setName method	Call the method with a list of	pass
09	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 300: Unit tests for pzmodel/setName.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/set	tOunits		
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the setOunits method	Test that the setOunits method	pass
-	works with a vector of	works for a vector of PZMODEL	
	PZMODEL objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'pzv' 2) Check that each output	
		contains the correct data.	
03	Tests that the setOunits method	Test that the setOunits method	pass
00	works with a matrix of	works for a matrix of PZMODEL	
	PZMODEL objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'pzm' 2) Check that each output	
		contains the correct data.	
04	Tests that the setOunits method	Test that the setOunits method	pass
01	works with a list of PZMODEL	works for a list of PZMODEL ob-	
	objects as input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		contains the correct data.	
05	Tests that the setOunits method	Test that the setOunits method	pass
	works with a mix of different	works with an input of matri-	
	shaped PZMODEL objects as	ces and vectors and single PZ-	
	input.	MODEL objects.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		contains the correct data.	
06	Tests that the setOunits method	Test that the result of apply-	pass
	properly applies history and	ing the setOunits method can be	
	that the option 'internal'	processed back to an m-file.	
	suppresses the history.	1) Check that the last entry in	pass
		the history of 'out1' corresponds	
		to setOunits'. 2) Check that the	
		NOT compare and to 'est Out'	
		2) Check that the method	
		5) Uneck that the method re-	
		build produces the same object	
		as 'out'.	
07	1 lests that the setOunits method	1 lest that the setOunits method	pass
	can modify the input	can modify the input PZMODEL	
	PZMODEL object.	object by calling with no output.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/setOunits			
		1) Check that 'pz5' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct ounits field	
08	Tests that the setOunits method	Test that the setOunits method	pass
00	can set the property with a	can modify the property 'ounits'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect ounits field 2) Check that	
		the method rebuild produces the	
		same object as 'out'.	
00	Check that the setOunits	Call the method with a list of	pass
09	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 301: Unit tests for pzmodel/setOunits.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

pzmodel/simplify			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the simplify method works with a vector of PZMODELs as input.	Test that the simplify method works for a vector of PZMOD- ELs as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'pzv' 2) Check the poles of the output 3) Check the zeros of the output	pass
03	Tests that the simplify method works with a matrix of PZMODELs as input.	Tests that the simplify method works with a matrix of PZMOD- ELs as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'pzm' 2) Check the poles of the output 3) Check the zeros of the output	pass
04	Tests that the simplify method works with a list of PZMODELs as input.	Tests that the simplify method works with a list of PZMODELs as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in the input 2) Check the poles of the output 3) Check the zeros of the output	pass
05	Tests that the simplify method works with a mix of different shaped PZMODELs as input.	Tests that the simplify method works with a mix of different shaped PZMODELs as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in the input 2) Check the poles of the output 3) Check the zeros of the output 4) Check the re-built object	pass
06	Tests that the simplify method properly applies history.	Test that the result of applying the simplify method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'simplify'. 2) Check that re- built object is the same object as the input.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/simplify			
07	Tests that the simplify method	Test that the simplify method	pass
01	can modify the input	can modify the input PZMODEL	
	PZMODEL.	by calling with no output and	
		that the method doesn't change	
		the input of the function nota-	
		tion (with a equal sign).	
		1) Check that 'out' and 'aeq' are	pass
		now different. 2) Check that 'aeq'	
		is not changed 3) Check that the	
		modified input is the simplified	
		value of the copy 4) Check that	
		out and amodi are the same	
0.0	Check that the simplify method	Check that the simplify method	pass
08	cancle poles/zeros at different	cancle poles/zeros at different	
	positions.	positions.	
		1) Check the poles and peros of	pass
		the output.	

Table 302: Unit tests for pzmodel/simplify.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/string			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the string method works with a vector of PZMODEL objects as input.	Test that the string method works for a vector of PZMODEL objects as input.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
03	Tests that the string method works with a matrix of PZMODEL objects as input.	Test that the string method works for a matrix of PZMODEL objects as input.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
04	Tests that the string method works with a list of PZMODEL objects as input.	Test that the string method works for a list of PZMODEL ob- jects as input.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
05	Tests that the string method works with a mix of different shaped PZMODEL objects as input.	Test that the string method works with an input of matri- ces and vectors and single PZ- MODEL objects.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
06	Tests that the string method properly applies history.	The method string doesn't change the data, thus it is not possible to check the history. Nothing to do.	pass
07	Tests that the string method doesn't work if the PZMODEL object have more than one history step	The method string throws an er- ror because the input object have more than one history step.	pass
	I motory such.		pass

Table 303: Unit tests for pzmodel/string.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/times			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the times method works with a vector of PZMODELs as input.	Test that the times method works for a vector of PZMOD- ELs as input.	pass
		1) Check that the output is exact one PZMODEL object. 2) Check the gain of the output 3) Check the poles of the output 4) Check the zeros of the output	pass
03	Tests that the times method works with a matrix of PZMODELs as input.	Tests that the times method works with a matrix of PZMOD- ELs as input.	pass
		1) Check that the output is exact one PZMODEL object. 2) Check the gain of the output 3) Check the poles of the output 4) Check the zeros of the output	pass
04	Tests that the times method works with a list of PZMODELs as input.	Tests that the times method works with a list of PZMODELs as input.	pass
		1) Check that the output is exact one PZMODEL object. 2) Check the gain of the output 3) Check the poles of the output 4) Check the zeros of the output	pass
05	Tests that the times method works with a mix of different shaped PZMODELs as input.	Tests that the times method works with a mix of different shaped PZMODELs as input.	pass
		1) Check that the output is exact one PZMODEL object. 2) Check the gain of the output 3) Check the poles of the output 4) Check the zeros of the output 5) Check the rebuilt object	pass
06	Tests that the times method properly applies history.	Test that the result of applying the times method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'times'. 2) Check that re-built object is the same object as the input.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/times			
07	Check that the times method	Check that the times method	pass
01	only multiply PZMODELs with	only multiply PZMODELs with	
	the same output units.	the same output units. Check	
		also the negative case.	
		1) Check the I-/O-units	pass

Table 304: Unit tests for pzmodel/times.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/tomfir			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the tomfir method	Test that the tomfir method	pass
02	works with a vector of	works for a vector of PZMODEL	
	PZMODEL objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'pzv' 2) Check that each output	
		PZMODEL object contains the	
		correct data.	
03	Tests that the tomfir method	Test that the tomfir method	pass
	works with a matrix of	works for a matrix of PZMODEL	
	PZMODEL objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'pzm' 2) Check that each output	
		PZMODEL object contains the	
		correct data.	
04	Tests that the tomfir method	Test that the tomfir method	pass
	works with a list of PZMODEL	works for a list of PZMODEL ob-	
	objects as input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each out-	
		put PZMODEL object contains	
		the correct data.	
05	Tests that the tomfir method	Test that the tomfir method	pass
	works with a mix of different	works with an input of matri-	
	shaped PZMODEL objects as	ces and vectors and single PZ-	
	input.	MODEL objects.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each out-	
		put PZMODEL object contains	
		the correct data.	
06	Tests that the tomfir method	Test that the result of applying	pass
	properly applies history.	the tomfir method can be pro-	
		cessed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to 'tomfir'. 2) Check that re-built	
		object is the same object as the	
		input.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

pzmodel/tomfir			
07	Tests that the tomfir method can modify the input PZMODEL object.	Test that the tomfir method can not modify the input PZMODEL object. The method must throw an error for the modifor call	pass
		1) Nothing to check.	pass
08	Control the method with a plist.	Test that the tomfir method use the different values in a plist.	pass
		1) Check the output 2) Check the re-built object	pass
09	Check that the tomfir method pass back the output objects to a list of output variables or to a single variable	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method	pass
	single variable.	works on the output.	
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass

Table 305: Unit tests for pzmodel/tomfir.


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/tomiir			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the tomiir method works with a vector of PZMODEL objects as input.	Test that the tomiir method works for a vector of PZMODEL objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'pzv'	pass
03	Tests that the tomiir method works with a matrix of PZMODEL objects as input.	Test that the tomiir method works for a matrix of PZMODEL objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'pzm'	pass
04	Tests that the tomiir method works with a list of PZMODEL objects as input.	Test that the tomiir method works for a list of PZMODEL ob- jects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each out- put PZMODEL object contains the correct data.	pass
05	Tests that the tomiir method works with a mix of different shaped PZMODEL objects as input.	Test that the tomiir method works with an input of matri- ces and vectors and single PZ- MODEL objects.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each out- put PZMODEL object contains the correct data.	pass
06	Tests that the tomiir method properly applies history.	Test that the result of applying the tomiir method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'tomiir'. 2) Check that re-built object is the same object as the input.	pass
07	Tests that the tomiir method can modify the input PZMODEL object.	Test that the tomiir method can not modify the input PZMODEL object . The method must throw an error for the modifier call.	pass
		1) Nothing to check.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

pzmodel/tomiir			
08	Control the method with a plict	Test that the tomiir method use	pass
08	Control the method with a plist.	the different values in a plist.	
		1) Check the output 2) Check the	pass
		re-built object	
00	Check that the tomiir method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
10	Check indirect the protected	Create some special pole/zero	pass
10	'pzm2ab' method because tomiir	models to test the 'pzm2ab'	
	uses this method to get the a	method indirect with the tomiir	
	and b of the iir object.	method. Test with complex	
		pole/zero pairs. This UTP	
		define a simple method to get	
		the a-, and b-value from a	
		complex pole/zero pair. It uses	
		the following code: function [a,b]	
		$=$ utp_cpz2ab(pf, pq, zf, zq, fs)	
		$wp = pf^*2^*pi; wp2 = wp2; wz =$	
		$zf^{*}2^{*}pi; wz2 = wz2; k = 4^{*}fs^{*}fs$	
		$+ 2^* wp^* fs/pq + wp2; a(1) =$	
		(4*fs*fs + 2*wz*fs/zq + wz2)/k;	
		$a(2) = (2^*wz2 - 8^*fs^*fs)/k;$	
		a(3) = (4*fs*fs - 2*wz*fs/zq +	
		wz2)/k; b(1) = 1; b(2) = (2*wp2)	
		- 8*fs*fs)/k; b(3) = (wp2 +	
		4*fs*fs - 2*wp*fs/pq)/k; g =	
		sum(a) / sum(b); a = a / g; end	
		1) Check the output 2) Check	pass
		that the 'rebuild' method pro-	
		duces the same object as 'out'.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



pzmodel/tomiir			
11	Check indirect the protected 'pzm2ab' method because tomiir uses this method to get the a and b of the iir object.	Create some special pole/zero models to test the 'pzm2ab' method indirect with the tomiir method. Test with complex poles. This UTPs define a simple method to get the a-, and b-value from a complex pole. It uses the following code: function [a,b] = utp_cp2iir(pf, pq, fs) w0 = pf*2*pi; w02 = w02; k = (pq*w02 + 4*pq*fs*fs + 2*w0*fs) / (pq*w02); b(1) = 1; b(2) = (2*w02-8*fs*fs) / (k*w02); b(3) = (pq*w02 + 4*pq*fs*fs - 2*w0*fs) / (k*pq*w02); a(1) = 1/k; a(2) = -2/k; a(3) = -1/k; a = a*-2; end	pass
		1) Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'.	pass
12	Check indirect the protected 'pzm2ab' method because tomiir uses this method to get the a and b of the iir object.	Create some special pole/zero models to test the 'pzm2ab' method indirect with the tomiir method. Test with complex ze- ros. This UTPs define a simple method to get the a-, and b- value from a complex zero. It uses the following code: func- tion $[a,b] = utp_cz2iir(zf, zq,$ fs) w0 = zf*2*pi; w02 = w02; a(1) = (-zq*w02/2 - 2*zq*fs*fs - w0*fs) / (zq*w02); a(2) = (- w02+4*fs*fs) / w02; a(3) = (- zq*w02/2 - 2*zq*fs*fs + w0*fs) / (zq*w02); b(1) = 1; b(2) = -2; b(3) = -1; end 1) Check the output 2) Check that the 'rebuild' method pro- duces the same object as 'out'.	pass
13	Check indirect the protected 'pzm2ab' method because tomiir uses this method to get the a and b of the iir object.	Create some special pole/zero models to test the 'pzm2ab' method indirect with the tomiir method. Test with real poles. This UTPs define a simple method to get the a-, and b- value from a real pole. It uses the following code: function $[a,b] =$ utp_rp2iir(pf, fs) w0 = pf*2*pi; a(1) = w0 / (2*fs + w0); a(2) = a(1); b(1) = 1; b(2) = (w0-2*fs) / (w0+2*fs); end	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/tomiir			
		1) Check the output 2) Check	pass
		that the 'rebuild' method pro-	
		duces the same object as 'out'.	
14	Check indirect the protected	Create some special pole/zero	pass
14	'pzm2ab' method because tomiir	models to test the 'pzm2ab'	
	uses this method to get the a	method indirect with the tomiir	
	and b of the iir object.	method. Test with real zeros.	
		This UTPs define a simple	
		method to get the a-, and b-	
		value from a real zero. It uses	
		the following code: function [a,b]	
		$=$ utp_rp2iir(zf, fs) w0 = zf*2*pi;	
		$a(1) = (2^{*}fs + w0) / w0; a(2) =$	
		(-2*fs + w0) / w0; b(1) = 1; b(2)	
		= 1; end	
		1) Check the output 2) Check	pass
		that the 'rebuild' method pro-	
		duces the same object as 'out'.	

Table 306: Unit tests for pzmodel/tomiir.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

pzmodel/type			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.0	Tests that the type method	Test that the type method works	pass
02	works with a vector of	for a vector of PZMODEL ob-	
	PZMODEL objects as input.	jects as input.	
		1) Check the rebuilt output.	pass
0.2	Tests that the type method	Test that the type method works	pass
05	works with a matrix of	for a matrix of PZMODEL ob-	
	PZMODEL objects as input.	jects as input.	
		1) Check the rebuilt output.	pass
0.4	Tests that the type method	Test that the type method works	pass
04	works with a list of PZMODEL	for a list of PZMODEL objects	
	objects as input.	as input.	
		1) Check the rebuilt output.	pass
05	Tests that the type method	Test that the type method works	pass
00	works with a mix of different	with an input of matrices and	
	shaped PZMODEL objects as	vectors and single PZMODEL	
	input.	objects.	
		1) Check the rebuilt output.	pass
06	Tests that the type method	The method type doesn't change	pass
00	properly applies history.	the data, thus it is not possible	
		to check the history. Nothing to	
		do.	
			pass

Table 307: Unit tests for pzmodel/type.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

rational/char			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set	pass
		1) Charle that wething and the	
		1) Check that getinio call re-	pass
		turned an minio object in an	
		cases. 2) Check that all plists	
	Tests that the shar mathed	Test that the chan mathed menta	
02	rests that the char method	for a motor of DATIONAL ab	pass
	DATIONAL shipsts as input	ior a vector of KATIONAL ob-	
	KATIONAL objects as input.	1) Check that the output contain	
		at least each object name	pass
03	Tests that the char method	Test that the char method works	pass
	works with a matrix of	for a matrix of RATIONAL ob-	
	RATIONAL objects as input.	jects as input.	
		1) Check that the output contain	pass
		at least each object name	
04	Tests that the char method	Test that the char method works	pass
01	works with a list of RATIONAL	for a list of RATIONAL objects	
	objects as input.	as input.	
		1) Check that the output contain	pass
		at least each object name	
05	Tests that the char method	Test that the char method works	pass
	works with a mix of different	with an input of matrices and	
	shaped RATIONAL objects as	vectors and single RATIONAL	
	input.	objects.	
		1) Check that the output contain	pass
		at least each object name	
06	Tests that the char method	The method char doesn't change	pass
	properly applies history.	the data, thus it is not possible	
		to check the history. Nothing to	
		do.	
			pass

Table 308: Unit tests for rational/char.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

rational/copy			
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	

Table 309: Unit tests for rational/copy.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



rational/created			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the created method works with a vector of RATIONAL objects as input.	Test that the created method works for a vector of RATIONAL objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'rav' 2) Check that each output contains the correct data.	pass
03	Tests that the created method works with a matrix of RATIONAL objects as input.	Test that the created method works for a matrix of RATIO- NAL objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'ram' 2) Check that each output contains the correct data.	pass
04	Tests that the created method works with a list of RATIONAL objects as input.	Test that the created method works for a list of RATIONAL objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the created method works with a mix of different shaped RATIONAL objects as input.	Test that the created method works with an input of matrices and vectors and single RATIO- NAL objects.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the created method properly applies history	This method doesn't change the input object, thus no history is added to the object.	pass
07	Tests that the created method can be used with the modify command.	Tests that the created method can be used with the modify command. 1) Check the single object 2)	pass pass pass
08	Tests that the created method retruns always a well defined time object even for an empty input object.	Check the matrix object Test that the created method with an empty 'RATIONAL ob- ject	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



rational/created		
	1) Check that the output is a	pass
	time object with a ell defined	
	time.	

Table 310: Unit tests for rational/created.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



rational/creator			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the creator method	Test that the creator method	pass
02	works with a vector of	works for a vector of RATIONAL	
	RATIONAL objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'rav' 2) Check that each output	
		contains the correct data.	
0.0	Tests that the creator method	Test that the creator method	pass
03	works with a matrix of	works for a matrix of RATIO-	
	RATIONAL objects as input.	NAL objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'ram' 2) Check that each output	
		contains the correct data.	
0.4	Tests that the creator method	The creator method doesn't work	pass
04	works with a list of RATIONAL	for a list of RATIONAL objects	1
	objects as input.	as input.	
		1) Nothing to test.	pass
~~~	Tests that the creator method	The creator method doesn't work	pass
05	works with a mix of different	with different shaped input ob-	
	shaped RATIONAL objects as	jects.	
	input.	1) Nothing to test	pass
	Tests that the creator method	This method doesn't change the	pass
06	properly applies history	input object, thus no history is	
		added to the object.	
		1) Nothing to check.	pass
	Tests that the creator method	Tests that the creator method	pass
07	can be used with the modify	can be used with the modify	1
	command.	command.	
		1) Check the single object 2)	pass
		Check the matrix object	
	Tests that the creator method	Test that the creator method	pass
08	retruns all	uses the option 'all' direct or in	
	creator(s)/modifier(s) which are	a plist. The test file must have	
	in the history.	the modifier 'first'. 'second' and	
		'third'	
		1) Check that out1 contains only	pass
		one creator 2) Check that out2	
		contain more creator/modifier	
	1		1



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



rational/creator			
09	Tests the negative case for the option 'all'.	Test that the creator method throws an error if the option 'all' is used in connection with a ma- trix/vector of RATIONAL ob- jects.	pass
		1) Nothing to test.	pass

Table 311: Unit tests for rational/creator.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

rational/display			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the display method	Test that the display method	pass
02	works with a vector of	works for a vector of RATIONAL	
	RATIONAL objects as input.	objects as input.	
		1) Check that the output contain	pass
		at least each object name	
03	Tests that the display method	Test that the display method	pass
00	works with a matrix of	works for a matrix of RATIO-	
	RATIONAL objects as input.	NAL objects as input.	
		1) Check that the output contain	pass
		at least each object name	
04	Tests that the display method	Test that the display method	pass
01	works with a list of RATIONAL	works for a list of RATIONAL	
	objects as input.	objects as input.	
		1) Check that the output contain	pass
		at least each object name	
05	Tests that the display method	Test that the display method	pass
00	works with a mix of different	works with an input of matrices	
	shaped RATIONAL objects as	and vectors and single RATIO-	
	input.	NAL objects as.	
		1) Check that the output contain	pass
		at least each object name	
06	Tests that the display method	The method display doesn't	pass
00	properly applies history.	change the data, thus it is not	
		possible to check the history.	
		Nothing to do.	
			pass

Table 312: Unit tests for rational/display.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

rational/eq			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the eq method works with a vector of RATIONAL objects as input.	Test that the eq method works for a vector of RATIONAL ob- jects as input. Test the positive and the negative case.	pass
03	Tests that the eq method works with a matrix of RATIONAL	function. Test that the eq method works for a matrix of RATIONAL ob-	pass
	objects as input.	<ul><li>and the negative case.</li><li>1) Check the output of the eq function.</li></ul>	pass
04	Tests that the eq method works with a list of RATIONAL objects as input.	The eq method doesn't works for a list of RATIONAL objects as input. Nothing to do.	pass
05	Tests that the eq method works with a mix of different shaped RATIONAL objects as input.	The eq method doesn't works for a list of RATIONAL objects as input. Nothing to do.	pass pass
06	Tests that the eq method properly applies history.	The eq method doesn't change the RATIONAL object, thus will no history added. Nothing to do	pass
07	Test the eq method with an exception list. With the LTPDA toolbox 2.0 it is only possible to test the exception list with properties where a public set method exist.	Test the eq method with the exception 'name'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because 'pa' is created at an other time. 1) Check the output.	pass pass
08	Test the eq method with an exception list. With the LTPDA toolbox 2.0 it is only possible to test the exception list with properties where a public set method exist.	Test the eq method with the exception 'iunits'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because 'pa' is created at an other time. 1) Check the output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

rational/eq			
00	Test the eq method with an	Test the eq method with the ex-	pass
	exception list. With the LTPDA	ception 'ounits'. Use the option	
	toolbox 2.0 it is only possible to	'internal' to suppress the history.	
	test the exception list with	It is necessary to add 'created' to	
	properties where a public set	the exception list because 'pa' is	
	method exist.	created at an other time.	
		1) Check the output.	pass
10	Test the eq method with an	Test that the eq method uses the	pass
10	exception list which is in a plist.	exception list in a plist.	
		1) Check the output.	pass

Table 313: Unit tests for rational/eq.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

rational/get			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests the get method of the	Test that the get returns returns	pass
02	rational class.	the value of the specified prop-	
		erty. Do this for all properties of	
		the RATIONAL object.	
		1) Check the correct value of the	pass
		output	
03	Tests that the get method works	Test that the get returns returns	pass
05	with a plist.	the value of the specified prop-	
		erty which is defined in a plist.	
		1) Check the correct value of the	pass
		output	
04	Tests the get method of the	Test that the get throws an error	pass
04	rational class.	if the input are more than one	
		RATIONAL object.	
		1) Nothing to test	pass

Table 314: Unit tests for rational/get.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

rational/getlowerI	req		
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests the getlowerFreq method	Test that the getlowerFreq re-	pass
02	of the rational class.	turns the lowest frequence of the	
		pole in the rational object.	
		1) Check the output	pass
02	Tests the getlowerFreq method	Test that the getlowerFreq	pass
05	of the rational class.	throws an error if the input are	
		more than one rational.	
		1) Nothing to test	pass

Table 315: Unit tests for rational/getlowerFreq.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

rational/getupper	Freq		
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests the getupperFreq method	Test that the getupperFreq re-	pass
02	of the rational class.	turns the lowest frequence in the	
		rational object.	
		1) Check the output	pass
02	Tests the getupperFreq method	Test that the getupperFreq	pass
03	of the rational class.	throws an error if the input are	
		more than one rational.	
		1) Nothing to test	pass

Table 316: Unit tests for rational/getupperFreq.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



rational/index			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set	pass
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the index method	Test that the index method	pass
02	works with a vector of	works for a vector of RATIONAL	
	RATIONAL objects as input.	objects as input. The following	
		indexing should work: $I = [1 2 3]$	
		] or $(I/J) = [(1,1), (1,2), (1,3)]$	
		1) Check that the index method	pass
		selects the correct object.	
03	Tests that the index method	Test that the index method	pass
	works with a matrix of	works for a matrix of RATIO-	
	RATIONAL objects as input.	NAL objects as input. The fol-	
		lowing indexing should work: $I = $	
		$\begin{bmatrix} 1 & 3 & 5 \end{bmatrix}$ or $(1/J) = [(1,1), (1,2), (1,2)]$	
		$\begin{array}{ c c c c c c c c c c c c c c c c c c c$	
		1) Check that the index method	pass
	Trata that the index mathed	The index wethed do en 't ment	
04	rests that the index method	for a list of PATIONAL abjects	pass
	works with a list of KATIONAL	ag input	
	objects as input.	1) Nothing to tost	nass
	Tests that the index method	Test that the result of index have	pass
05	properly applies history	an additional history step	pass
	property applies instory.	1) Check that the last entry in	nass
		the history of 'out' corresponds	Pabb
		to 'index'.	
	Tests that the index method	Tests that the index method	pass
06	works for the modifier	works for the modifier command.	
	command.	1) Check that the history-plist	pass
		contains the used indices. 2)	
		Check that the index method se-	
		lects the correct object	
07	Control the method with a plist	Test that the index method can	pass
01	Control the method with a plist.	be controled with a plist.	
		1) Check that the history-plist	pass
		contains the used indices. 2)	
		Check that the index method se-	
		lects the correct object	
08	Test that the index method	Test that the index method se-	pass
	selects more objects if I have	lects more objects if I have more	
	more indices.	indices.	
		1) Check that the history-plist	pass
		contains the used indices. 2)	
		Check that the index method se-	
		lects the correct object	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



rational/index
----------------

Table 317: Unit tests for rational/index.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



rational/isprop			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the isprop method works with a vector of RATIONAL objects as input.	Test that the isprop method works for a vector of RATIONAL objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'rav' 2) Check that each output contains the correct data.	pass
03	Tests that the isprop method works with a matrix of RATIONAL objects as input.	Test that the isprop method works for a matrix of RATIO- NAL objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'ram' 2) Check that each output contains the correct data.	pass
04	Tests that the isprop method works with a list of RATIONAL objects as input.	Test that the isprop method works for a list of RATIONAL objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the isprop method works with a mix of different shaped RATIONAL objects as input.	Test that the isprop method works with an input of matrices and vectors and single RATIO- NAL objects.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the isprop method properly applies history.	The method isprop doesn't change the object, thus it is not necessary to apply history.	pass
07	Tests that the isprop method works for each property.	Test that the isprop method works for the properties: 'num', 'den', 'iunits', 'ounits', 'hist', 'name'	pass
		1) Check that each output con- tains the correct data.	pass
08	Test the negative case and the not function command.	Test that the isprop method retrun false for a unknown prop- erty and for methods of the ob- ject.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



rational/isprop			
	1) Check that each output	t con-	pass
	tains the correct data.		

Table 318: Unit tests for rational/isprop.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



rational/loadobj			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check the shape of the loaded	pass
		objects.	

Table 319: Unit tests for rational/loadobj.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

rational/ne			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the ne method works with a vector of RATIONAL objects as input.	Test that the ne method works for a vector of RATIONAL ob- jects as input. Test the positive and the negative case.	pass
		1) Check the output of the ne function.	pass
03	Tests that the ne method works with a matrix of RATIONAL objects as input.	Test that the ne method works for a matrix of RATIONAL ob- jects as input. Test the positive and the negative case.	pass
		1) Check the output of the ne function.	pass
04	Tests that the ne method works with a list of RATIONAL objects as input.	The ne method doesn't works for a list of RATIONAL objects as input. Nothing to do.	pass
05	Tests that the ne method works with a mix of different shaped RATIONAL objects as input.	The ne method doesn't works for a list of RATIONAL objects as input. Nothing to do.	pass
			pass
06	Tests that the ne method properly applies history.	The ne method doesn't change the RATIONAL object, thus will no history added. Nothing to do	pass
	That the mean the density and	The static second state of second states and	pass
07	rest the ne method with an exception list. The function rational/ne use the function rational/eq so it is not necessary to check all possibilities of the exception list.	rest the ne method with the ex- ception 'name'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because ra is created at an other time.	pass
		1) Uneck that each output con-	pass
08	Test the ne method with an exception list which is in a plict	Test that the ne method uses the exception list in a plist	pass
	exception list which is in a plist.	1) Check that each output con- tains the correct data.	pass

Table 320: Unit tests for rational/ne.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



rational/rational			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the rational method	Test that the rational method	pass
02	works with a vector of	works with a vector of RATIO-	
	RATIONAL objects as input.	NAL objects as input.	
		1) Check that the shape of the	pass
		output RATIONALs is the same	
		as the input shape. 2) Check	
		that each output RATIONAL is	
		a copy of the input RATIONAL.	
		3) Check that the copy have an	
		additional history step.	
03	Tests that the rational method	Test that the rational method	pass
00	works with a matrix of	works with a matrix of RATIO-	
	RATIONAL objects as input.	NAL objects as input.	
		1) Check that the shape of the	pass
		output RATIONALs is the same	
		as the input shape. 2) Check	
		that each output RATIONAL is	
		a copy of the input RATIONAL.	
		3) Check that the copy have an	
		additional history step.	
04	Tests that the rational method	Test that the rational method	pass
L L L L L L L L L L L L L L L L L L L	works with a list of RATIONAL	works with a list of RATIONAL	
	objects as input.	objects as input.	
		1) Check that the number of ele-	pass
		ments in 'out' is the same of the	
		number in the input. 2) Check	
		that each output RATIONAL is	
		a copy of the input RATIONAL.	
		3) Check that the copy have an	
		additional history step.	
05	Tests that the rational method	Test that the rational method	pass
00	works with a mix of different	works with a mix of different	
	shaped RATIONALs as input.	shaped RATIONALs as input.	
		1) Check that the number of ele-	pass
		ments in 'out' is the same of the	
		number in the input. 2) Check	
		that each output RATIONAL is	
		a copy of the input RATIONAL.	
		3) Check that the copy have an	
		additional history step.	
06	Tests that the rational method	Test that the result of applying	pass
	properly applies history.	the rational method can be pro-	
		cessed back.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



## università degli studi di trento

rational/rational			
	Tests that the rational method	1) Check that the last entry in the history of 'out' corre- sponds to 'rational'. 2) Check that the method rebuild pro- duces the same object as 'out'.	pass
07	properly applies history to the copy constructor.	cessed back with the 'rebuild' method. Test the constructor with a different number of in- puts.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'rational'. 2) Check that the original objects are not changed by the setter function 3) Check that the method rebuild pro- duces the same object as 'out'.	pass
08	Tests that the rational method properly applies history to the read MAT-file constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
09	Tests that the rational method properly applies history to the read XML-file constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		method produces the same object as 'out'.	pass
10	Tests that the rational method properly doesn't apply history to the struct constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'rational'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
11	Tests that the rational method properly applies history to the parfrac constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'rational'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
12	Tests that the rational method properly applies history to the pzmodel constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



### università degli studi di trento

rational/rational			
		1) Check that the last entry in the history of 'out' corresponds to 'rational'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
13	Tests that the rational method properly applies history to the	Test that the output can be pro- cessed back to an m-file.	pass
	plist(filename) constructor.	1) Check that the save method doesn't change the input ob- ject 2) Check that the last two entries in the history of 'out' doesn't corresponds to 'rational' and 'save' 3) Check that the 're- build' method produces the same object as 'out'.	pass
14	Tests that the RATIONAL method properly applies history to the plist(conn) constructor.	Test that the output can be processed back with the rebuild method.	pass
		the history of 'out' corresponds to 'rational'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
15	Tests that the RATIONAL method properly applies history to the plist(den    num)	Test that the output can be processed back with the rebuild method.	pass
	constructor.	1) Check that the last entry in the history of 'out' corresponds to 'rational'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
16	Tests that the RATIONAL method properly applies history to the plist(pzmodel)	Test that the output can be processed back with the rebuild method.	pass
	constructor.	1) Check that the last entry in the history of 'out' corresponds to 'rational'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
17	Tests that the RATIONAL method properly applies history to the plist(rational)	Test that the output can be processed back with the rebuild method.	pass
	constructor.	1) Check that the last entry in the history of 'out' corresponds to 'rational'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
18	Tests that the RATIONAL method properly applies history to the plist( <plist-object>) constructor</plist-object>	Test that the output can be processed back with the rebuild method.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



#### UNIVERSITÀ DEGLI STUDI DI TRENTO

rational/rational			
	Tests that the RATIONAL	1) Check that the last entry in the history of 'out' corresponds to 'rational'. 2) Check that the 'rebuild' method produces the same object as 'out'. Test that the output can be	pass
19	method properly applies history to the conn+Id constructor.	processed back with the rebuild method.	Pass
		1) Check that the last entry in the history of 'out' corresponds to 'rational'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
20	Tests that the RATIONAL method properly applies history to the num + den object	Test that the output can be processed back with the rebuild method.	pass
	constructor.	1) Check that the last entry in the history of 'out' corresponds to 'rational'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
21	Tests that the RATIONAL method properly applies history to the num + den + name	Test that the output can be processed back with the rebuild method.	pass
	object constructor.	1) Check that the last entry in the history of 'out' corresponds to 'rational'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
22	Tests that the RATIONAL method properly applies history to the num + den + name +	Test that the output can be processed back with the rebuild method.	pass
	iunits + ounits object constructor.	1) Check that the last entry in the history of 'out' corresponds to 'rational'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass

Table 321: Unit tests for rational/rational.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

rational/rebuild			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the rebuild method	Test that the rebuild method	pass
02	works with a vector of	works for a vector of RATIONAL	
	RATIONAL objects as input.	objects as input.	
		1) Check the rebuilt output.	pass
03	Tests that the rebuild method	Test that the rebuild method	pass
00	works with a matrix of	works for a matrix of RATIO-	
	RATIONAL objects as input.	NAL objects as input.	
		1) Check the rebuilt output.	pass
04	Tests that the rebuild method	Test that the rebuild method	pass
04	works with a list of RATIONAL	works for a list of RATIONAL	
	objects as input.	objects as input.	
		1) Check the rebuilt output.	pass
05	Tests that the rebuild method	Test that the rebuild method	pass
00	works with a mix of different	works with an input of matrices	
	shaped RATIONAL objects as	and vectors and single RATIO-	
	input.	NAL objects.	
		1) Check the rebuilt output.	pass
06	Tests that the rebuild method	The method rebuild doesn't	pass
	properly applies history.	change the data, thus it is not	
		possible to check the history.	
		Nothing to do.	
			pass
07	Check that the rebuild method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 322: Unit tests for rational/rebuild.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



rational/resp			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the resp method	Test that the resp method works	pass
02	works with a vector of	for a vector of RATIONAL ob-	
	RATIONAL objects as input.	jects as input. Test the method	
		with an output and with no out-	
		put (a diagram must appear)	
		1) Test the right number of lines	pass
		in the diagram. 2) Check that the	-
		number of elements in 'out' is the	
		same as in 'rav' 3) Check that	
		each output RATIONAL con-	
		tains the correct data.	
0.2	Tests that the resp method	Tests that the resp method works	pass
03	works with a matrix of	with a matrix of RATIONAL ob-	-
	RATIONAL objects as input.	jects as input. Test the method	
		with an output and with no out-	
		put (a diagram must appear)	
		1) Test the right number of lines	pass
		in the diagram. 2) Check that the	
		number of elements in 'out' is the	
		same as in 'ram' 3) Check that	
		each output RATIONAL con-	
		tains the correct data.	
0.4	Tests that the resp method	Tests that the resp method works	pass
04	works with a list of RATIONAL	with a list of RATIONAL objects	
	objects as input.	as input. Test the method with	
		an output and with no output (a	
		diagram must appear)	
		1) Test the right number of lines	pass
		in the diagram. 2) Check that the	
		number of elements in 'out' is the	
		same as in 'rain' 3) Check that	
		each output RATIONAL con-	
		tains the correct data.	
05	Tests that the resp method	Tests that the resp method works	pass
	works with a mix of different	with a mix of different shaped	
	shaped RATIONAL objects as	RATIONAL objects as input.	
	input.	Test the method with an output	
		and with no output (a diagram	
		must appear)	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



#### università degli studi di trento

rational/resp			
		1) Test the right number of lines in the diagram. 2) Check that the number of elements in 'out' is the same as in 'rain' 3) Check that each output RATIONAL con- tains the correct data.	pass
06	properly applies history.	Test that the result of apply- ing the resp method can be pro- cessed back.	pass
		the history of 'out' corresponds to 'resp'. 2) Check that re-built object is the same object as the input.	Pass
07	Tests that modify command plots the response into a diagram.	Tests that modify command plots the response into a dia- gram.	pass
08	Test the shape of the output.	Test that the output AO of the resp method keeps the shape of the used input f vector	pass
		<ol> <li>the used input I vector.</li> <li>Check that the shape of the data doesn't change.</li> </ol>	pass
09	Check that the resp method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
10	Check that the resp method uses the x-data of an input AO for f-vector.	Call the method with different method to pass an AO in. 1) Check the output 2) Check that the 'rebuild' method pro-	pass pass
		duces the same object as 'out'.	
11	Check that the resp method uses the specified f-vector to compute the response	Call the method with different method to pass an f-vector in.	pass
		that the 'rebuild' method pro- duces the same object as 'out'.	Pass
12	Check that the resp method uses the specified f1, f2, and nf to compute the response	Call the method with different method to pass f1, f2, and nf in. 1) Check the output 2) Check	pass
	to compute the response.	that the 'rebuild' method pro- duces the same object as 'out'.	pass
13	Check that the resp method uses the specified f1, f2, and nf to compute the response.	Call the method with different method to pass f1, f2, and nf in.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



rational/resp		
	1) Check the output 2) Check	pass
	that the 'rebuild' method pro-	
	duces the same object as 'out'.	

Table 323: Unit tests for rational/resp.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



rational/save			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the save method works with a vector of RATIONAL objects as input.	Test that the save method works for a vector of RATIONAL ob- jects as input. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out1' and 'out2' are the same as in 'rav' 2) Check that the loaded objects are the same as the saved objects. 3) The out- puts 'out1' and 'out2' must be the same.	pass
03	Tests that the save method works with a matrix of RATIONAL objects as input.	Test that the save method works for a matrix of RATIONAL ob- jects as input. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of ele- ments in 'out1' and 'out2' are the same as in 'ram' 2) Check that the loaded objects are the same as the saved objects. 3) The out- puts 'out1' and 'out2' must be the same.	pass
04	Tests that the save method works with a list of RATIONAL objects as input.	Test that the save method works for a list of RATIONAL objects as input. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out1' and 'out2' are the same as in the list 2) Check that the loaded objects are the same as the saved objects. 3) The outputs 'out1' and 'out2' must be the same.	pass
05	Tests that the save method works with a mix of different shaped RATIONAL objects as input.	Test that the save method works with an input of matrices and vectors and single RATIONAL objects. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output RATIONAL object contains the correct data.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

rational/save			
06	Tests that the save method	Test that the result of apply-	pass
00	properly applies history.	ing the save method can be pro-	
		cessed back to an m-file. Do this	
		for both extensions 'mat' and	
		'xml'	
		1) Check that the history applies	pass
		to the output object. Check that	
		save doesn't add a history step to	
		the input object. 2) Check that	
		the read object doesn't contain	
		the save + load history steps.	
		3) Check that the method re-	
		build produces the same object	
		as 'out'.	
07	Tests that the save method	Use the save method with the	pass
	works with the modify	modifier command.	
	command.	1) Check that the save method	pass
		doesn't apply the history. 2)	
		Check the output against the in-	
		put.	
08	Control the method with a plist	Test that the save method uses	pass
00	Control the method with a plist.	the filename which is stored in a	
		plist.	
		1) Check the output	pass
00	Test the save method with	Test the save method with differ-	pass
	different complex RATIONAL	ent complex RATIONAL objects	
	objects	1) Check the output	pass

Table	324:	Unit	tests	for	rational	/save.
10010	0	0 1110	00000	101	rautonar	10010.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



rational/setIunits			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the setIunits method	Test that the setIunits method	pass
02	works with a vector of	works for a vector of RATIONAL	
	RATIONAL objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'rav' 2) Check that each output	
		contains the correct data.	
0.2	Tests that the setIunits method	Test that the setIunits method	pass
03	works with a matrix of	works for a matrix of RATIO-	-
	RATIONAL objects as input.	NAL objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	_
		'ram' 2) Check that each output	
		contains the correct data.	
0.4	Tests that the setIunits method	Test that the setIunits method	pass
04	works with a list of RATIONAL	works for a list of RATIONAL	-
	objects as input.	objects as input.	
	-	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		contains the correct data.	
05	Tests that the setIunits method	Test that the setIunits method	pass
05	works with a mix of different	works with an input of matrices	_
	shaped RATIONAL objects as	and vectors and single RATIO-	
	input.	NAL objects.	
	-	1) Check that the number of el-	pass
		ements in 'out' is the same as in	_
		input. 2) Check that each output	
		contains the correct data.	
0.0	Tests that the setIunits method	Test that the result of applying	pass
00	properly applies history and	the setIunits method can be pro-	
	that the option 'internal'	cessed back to an m-file.	
	suppresses the history.	1) Check that the last entry in	pass
		the history of 'out1' corresponds	
		to 'setIunits'. 2) Check that the	
		last entry in the history of 'out2'	
		NOT corresponds to 'setIunits'.	
		3) Check that the method re-	
		build produces the same object	
		as 'out'.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

rational/setIunits			
07	Tests that the setIunits method	Test that the setIunits method	pass
	can modify the input	can modify the input RATIO-	
	RATIONAL object.	NAL object by calling with no	
		output.	
		1) Check that 'ra3' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct iunits field	
08	Tests that the setIunits method	Test that the setIunits method	pass
00	can set the property with a	can modify the property 'iunits'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect iunits field 2) Check that	
		the method rebuild produces the	
		same object as 'out'.	
00	Check that the setIunits method	Call the method with a list of	pass
09	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 325: Unit tests for rational/setIunits.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



rational/setName			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
	Tests that the setName method	Test that the setName method	pass
02	works with a vector of	works for a vector of RATIONAL	
	RATIONAL objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	P
		'ray' 2) Check that each output	
		contains the correct data	
	Tests that the setName method	Test that the setName method	nass
03	works with a matrix of	works for a matrix of BATIO-	pass
	RATIONAL objects as input	NAL objects as input	
		1) Check that the number of el-	nase
		omonts in 'out' is the same as in	pass
		'ram' 2) Check that each output	
		contains the correct data	
	Tests that the setName method	Test that the setName method	naga
04	works with a list of <b>PATIONAL</b>	works for a list of <b>PATIONAL</b>	pass
	objects as input	objects as input	
	objects as input.	1) Check that the number of al	nagg
		amonts in 'out' is the same as in	pass
		input 2) Check that each output	
		approximate a construction output	
	Trata that the act Name mathed	The that the set News wethed	
05	rests that the setName method	Test that the setName method	pass
	works with a mix of different	works with an input of matrices	
	inner	NAL abiasta	
	input.	NAL objects.	
		1) Check that the number of el-	pass
		import 2) Charle that as the surface	
		mput. 2) Cneck that each output	
		contains the correct data.	
06	Tests that the setName method	Test that the result of applying	pass
	properly applies history and	the setName method can be pro-	
	that the option 'internal'	cessed back to an m-file.	
	suppresses the history.	1) Check that the last entry in	pass
		the history of 'outl' corresponds	
		to 'setName'. 2) Check that the	
		last entry in the history of 'out2'	
		NOT corresponds to 'setName'.	
		3) Check that the method re-	
		build produces the same object	
		as 'out'.	


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

rational/setName			
07	Tests that the setName method	Test that the setName method	pass
	can modify the input	can modify the input RATIO-	
	RATIONAL object.	NAL object by calling with no	
		output.	
		1) Check that 'ra3' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct name field	
08	Tests that the setName method	Test that the setName method	pass
00	can set the property with a	can modify the property 'name'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect name field 2) Check that	
		the method rebuild produces the	
		same object as 'out'.	
00	Check that the setName method	Call the method with a list of	pass
03	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 326: Unit tests for rational/setName.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



rational/setOun	its		
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
00	Tests that the setOunits method	Test that the setOunits method	pass
02	works with a vector of	works for a vector of RATIONAL	
	RATIONAL objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	1
		'ray' 2) Check that each output	
		contains the correct data.	
	Tests that the setOunits method	Test that the setOunits method	pass
03	works with a matrix of	works for a matrix of RATIO-	P
	RATIONAL objects as input.	NAL objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	P
		'ram' 2) Check that each output	
		contains the correct data.	
	Tests that the setOunits method	Test that the setOunits method	nass
04	works with a list of <b>BATIONAL</b>	works for a list of BATIONAL	Pabb
	objects as input.	objects as input.	
	osjeeus as mpau	1) Check that the number of el-	pass
		ements in 'out' is the same as in	P
		input, 2) Check that each output	
		contains the correct data.	
	Tests that the setOunits method	Test that the setOunits method	pass
05	works with a mix of different	works with an input of matrices	P
	shaped RATIONAL objects as	and vectors and single RATIO-	
	input.	NAL objects.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	P
		input, 2) Check that each output	
		contains the correct data.	
	Tests that the setOunits method	Test that the result of apply-	pass
06	properly applies history and	ing the setOunits method can be	Pass
	that the option 'internal'	processed back to an m-file	
	suppresses the history	1) Check that the last entry in	pass
	- approved the instory.	the history of 'out1' corresponds	1 1 2 2 2 2 2
		to 'setOunits', 2) Check that the	
		last entry in the history of 'out?'	
		NOT corresponds to 'setOunits'	
		3) Check that the method ro	
		build produces the same object	
		s out?	
1		as out.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

rational/setOunits			
07	Tests that the setOunits method	Test that the setOunits method	pass
	can modify the input	can modify the input RATIO-	
	RATIONAL object.	NAL object by calling with no	
		output.	
		1) Check that 'ra3' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct ounits field	
08	Tests that the setOunits method	Test that the setOunits method	pass
00	can set the property with a	can modify the property 'ounits'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect ounits field 2) Check that	
		the method rebuild produces the	
		same object as 'out'.	
00	Check that the setOunits	Call the method with a list of	pass
09	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 327: Unit tests for rational/setOunits.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

rational/string			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the string method works with a vector of RATIONAL objects as input.	Test that the string method works for a vector of RATIONAL objects as input.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
03	Tests that the string method works with a matrix of RATIONAL objects as input.	Test that the string method works for a matrix of RATIO- NAL objects as input.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
04	Tests that the string method works with a list of RATIONAL objects as input.	Test that the string method works for a list of RATIONAL objects as input.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
05	Tests that the string method works with a mix of different shaped RATIONAL objects as input.	Test that the string method works with an input of matrices and vectors and single RATIO- NAL objects.	pass
		1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
06	Tests that the string method properly applies history.	The method string doesn't change the data, thus it is not possible to check the history. Nothing to do.	pass
07	Tests that the string method doesn't work if the RATIONAL object have more than one history step.	The method string throws an er- ror because the input object have more than one history step.	pass

Table 328: Unit tests for rational/string.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

rational/type			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.9	Tests that the type method	Test that the type method works	pass
02	works with a vector of	for a vector of RATIONAL ob-	
	RATIONAL objects as input.	jects as input.	
		1) Check the rebuilt output.	pass
02	Tests that the type method	Test that the type method works	pass
05	works with a matrix of	for a matrix of RATIONAL ob-	
	RATIONAL objects as input.	jects as input.	
		1) Check the rebuilt output.	pass
04	Tests that the type method	Test that the type method works	pass
04	works with a list of RATIONAL	for a list of RATIONAL objects	
	objects as input.	as input.	
		1) Check the rebuilt output.	pass
05	Tests that the type method	Test that the type method works	pass
00	works with a mix of different	with an input of matrices and	
	shaped RATIONAL objects as	vectors and single RATIONAL	
	input.	objects.	
		1) Check the rebuilt output.	pass
06	Tests that the type method	The method type doesn't change	pass
00	properly applies history.	the data, thus it is not possible	
		to check the history. Nothing to	
		do.	
			pass

Table 329: Unit tests for rational/type.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



smodel/addAliase	\$		
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
genericAnvShape			pass
0			pass
genericAnvShape			pass
0			pass
genericAnvShape			pass
0			pass
genericAnyShape			pass
Senerici ing Sinapo			pass
genericAnyShape			pass
Semerieringsmape			pass
genericAnyShape			pass
0			pass
genericList	Tests that the	Tests that the	pass
Serrerrerrer	[smodel/addAliases] method	[smodel/addAliases] method	
	works for a list of objects as	works for a list of objects as	
	input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'objs' 2) Check that each output	
		object contains the correct data.	
genericHistory	Tests that the	Test that the result of applying	pass
8	[smodel/addAliases] method	the [smodel/addAliases] method	
	properly applies history.	can be processed back.	
		1) Check that the last entry	pass
		in the history of 'out' corre-	
		sponds to '[smodel/addAliases]'.	
		2) Check that the re-built object	
		is the same object as the input.	
genericModify	Tests that the	Test that the	pass
	[smodel/addAliases] method can	[smodel/addAliases] method	
	modify the input AO.	can modify the input object by	
		calling with no output and that	
		the method doesn't change the	
		input of the function notation	
		(with a equal sign).	
		1) Check that the modified in-	pass
		put is changed by the method	
		2) Check that 'out' and 'obj_eq'	
		are now different. 3) Check that	
		obj_eq' is not changed 4) Check	
		that out and amodi are the same	

genericOutput

smodel/addAliases



MAX PLANCK INSTITUTE FOR GRAVITATIONAL PHYSICS (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO Check that the Call the method with a list of pass [smodel/addAliases] method pass back the output objects output variables and with a sin-gle output variable. Additionally

pass back the output objects to	gle output variable. Additionaly	
a list of output variables or to a	check that the rebuild method	
single variable.	works on the output.	
	1) Check that the output con-	pass
	tains the right number of ob-	
	jects 2) Check that the 'rebuild'	
	method produces the same ob-	
	ject as 'out'.	

Table 330: Unit tests for smodel/addAliases.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



smodel/addParan	neters		
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
genericAnyShape	Tests that the [smodel/addParameters] method works with any shape of objects as input.	Test that the [smodel/addParameters] method works for any shape of objects as input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericAnyShape	Tests that the [smodel/addParameters] method works with any shape of objects as input.	Testthatthe[smodel/addParameters] methodworks for any shape of objectsas input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericAnyShape	Tests that the [smodel/addParameters] method works with any shape of objects as input.	Test that the [smodel/addParameters] method works for any shape of objects as input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericAnyShape	Tests that the [smodel/addParameters] method works with any shape of objects as input.	Test that the [smodel/addParameters] method works for any shape of objects as input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericAnyShape	Tests that the [smodel/addParameters] method works with any shape of objects as input.	Test that the [smodel/addParameters] method works for any shape of objects as input.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



# università degli studi di trento

smodel/addParan	neters		
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericAnyShape	Tests that the [smodel/addParameters] method works with any shape of objects as input.	Test that the [smodel/addParameters] method works for any shape of objects as input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericAnyShape	Tests that the [smodel/addParameters] method works with any shape of objects as input.	Test that the [smodel/addParameters] method works for any shape of objects as input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericAnyShape	Tests that the [smodel/addParameters] method works with any shape of objects as input.	Test that the [smodel/addParameters] method works for any shape of objects as input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericModify	Tests that the [smodel/addParameters] method can modify the input AO.	Test that the [smodel/addParameters] method can modify the input object by calling with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass
		1) Check that the modified in- put is changed by the method 2) Check that 'out' and 'obj_eq' are now different. 3) Check that 'obj_eq' is not changed 4) Check that out and amodi are the same	pass
genericOutput	Check that the [smodel/addParameters] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

smodel/addParan	neters		
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
genericAnyShape			pass
			pass
genericAnyShape			pass
	Tests that the	Tests that the	pass
genericList	[smodel/addParameters] method works for a list of objects as input.	<ul> <li>1 (smodel/addParameters] method works for a list of objects as input.</li> <li>1) Check that the number of elements in 'out' is the same as in</li> </ul>	pass
		'objs' 2) Check that each output object contains the correct data.	
genericHistory	Tests that the [smodel/addParameters] method properly applies history.	Test that the result of apply- ing the [smodel/addParameters] method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[smodel/addParameters]'. 2) Check that the re-built object is the same object as the input.	pass

Table 331: Unit tests for smodel/addParameters.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



smodel/clearAlias	ses		
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set	pass
		individually.1) Check that getInfo call re-turned an minfo object in allcases. 2) Check that all plistshave the correct parameters.	pass
genericAnyShape	Tests that the [smodel/clearAliases] method works with any shape of objects as input.	Testthatthe[smodel/clearAliases]methodworks for any shape of objectsas input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericAnyShape			pass
			pass
genericAnyShape			pass
genericList	Tests that the [smodel/clearAliases] method works for a list of objects as input.	Teststhatthe[smodel/clearAliases]methodworksfor a list of objects asinput.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericHistory	Tests that the [smodel/clearAliases] method properly applies history.	Test that the result of ap- plying the [smodel/clearAliases] method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[smodel/clearAliases]'. 2) Check that the re-built object is the same object as the input.	pass
genericModify	Tests that the [smodel/clearAliases] method can modify the input AO.	Test that the [smodel/clearAliases] method can modify the input object by calling with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass
		1) Check that the modified in- put is changed by the method 2) Check that 'out' and 'obj_eq' are now different. 3) Check that 'obj_eq' is not changed 4) Check that out and amodi are the same	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



smodel/clearAliases Check that the Call the method with a list of pass genericOutput[smodel/clearAliases] method output variables and with a single output variable. Additionaly pass back the output objects to a list of output variables or to a check that the rebuild method single variable. works on the output. 1) Check that the output conpass tains the right number of objects 2) Check that the 'rebuild' method produces the same object as 'out'.

Table 332: Unit tests for smodel/clearAliases.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

smodel/copy			
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	

Table 333: Unit tests for smodel/copy.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



smodel/loadobj			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check the shape of the loaded	pass
		objects.	

Table 334: Unit tests for smodel/loadobj.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)





Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

smodel/setAliases			
gonoriaOutput	Check that the	Call the method with a list of	pass
genericOutput	[smodel/setAliases] method pass	output variables and with a sin-	
	back the output objects to a list	gle output variable. Additionaly	
	of output variables or to a single	check that the rebuild method	
	variable.	works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 335: Unit tests for smodel/setAliases.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



smodel/setParame	eters		
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
genericAnyShape	Tests that the [smodel/setParameters] method works with any shape of objects as input.	Test that the [smodel/setParameters] method works for any shape of objects as input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericAnyShape	Tests that the [smodel/setParameters] method works with any shape of objects as input.	Test that the [smodel/setParameters] method works for any shape of objects as input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericAnyShape	Tests that the [smodel/setParameters] method works with any shape of objects as input.	Testthatthe[smodel/setParameters]methodworks for any shape of objectsas input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericAnyShape	Tests that the [smodel/setParameters] method works with any shape of objects as input.	Testthatthe[smodel/setParameters]methodworks for any shape of objectsas input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericAnyShape	Tests that the [smodel/setParameters] method works with any shape of objects as input.	Testthatthe[smodel/setParameters]methodworks for any shape of objectsas input.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



smodel/setParame	eters		
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericAnyShape	Tests that the [smodel/setParameters] method works with any shape of objects as input.	Test that the [smodel/setParameters] method works for any shape of objects as input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericAnyShape	Tests that the [smodel/setParameters] method works with any shape of objects as input.	Test that the [smodel/setParameters] method works for any shape of objects as input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericAnyShape	Tests that the [smodel/setParameters] method works with any shape of objects as input.	Test that the [smodel/setParameters] method works for any shape of objects as input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericModify	Tests that the [smodel/setParameters] method can modify the input AO.	Test that the [smodel/setParameters] method can modify the input object by calling with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass
		1) Check that the modified in- put is changed by the method 2) Check that 'out' and 'obj_eq' are now different. 3) Check that 'obj_eq' is not changed 4) Check that out and amodi are the same	pass
genericOutput	Check that the [smodel/setParameters] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

smodel/setParam	eters		
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
genericAnyShape			pass
man ani a A marCh an a			pass
genericAnyShape			pass
genericList	Tests that the [smodel/setParameters] method works for a list of objects as input.	Teststhatthe[smodel/setParameters]methodworks for a list of objects asinput.1)Check that the number of el-ements in 'out' is the same as in'objs' 2)Check that each outputobject contains the correct data.	pass
genericHistory	Tests that the [smodel/setParameters] method properly applies history.	Test that the result of apply- ing the [smodel/setParameters] method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[smodel/setParameters]'. 2) Check that the re-built object is the same object as the input.	pass

Table 336: Unit tests for smodel/setParameters.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



smodel/setParams			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
genericAnyShape	Tests that the [smodel/setParams] method works with any shape of objects as input.	Testthatthe[smodel/setParams]methodworks for any shape of objectsas input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericAnyShape	Tests that the [smodel/setParams] method works with any shape of objects as input.	Test that the [smodel/setParams] method works for any shape of objects as input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericAnyShape	Tests that the [smodel/setParams] method works with any shape of objects as input.	Testthatthe[smodel/setParams]methodworks for any shape of objectsas input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericAnyShape	Tests that the [smodel/setParams] method works with any shape of objects as input.	Testthatthe[smodel/setParams]methodworks for any shape of objectsas input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.</li> <li>Rebuild the object</li> </ol>	pass
genericAnyShape	Tests that the [smodel/setParams] method works with any shape of objects as input.	Testthatthe[smodel/setParams]methodworks for any shape of objectsas input.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



#### università degli studi di trento

smodel/setParams	s		
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'objs' 2) Check that each output	
		object contains the correct data.	
		3) Rebuild the object	
generic Any Shape	Tests that the	Test that the	pass
generierinysnape	[smodel/setParams] method	[smodel/setParams] method	
	works with any shape of objects	works for any shape of objects	
	as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'objs' 2) Check that each output	
		object contains the correct data.	
		3) Rebuild the object	
generic AnyShape	Tests that the	Test that the	pass
Senerieringsmape	[smodel/setParams] method	[smodel/setParams] method	
	works with any shape of objects	works for any shape of objects	
	as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'objs' 2) Check that each output	
		object contains the correct data.	
		3) Rebuild the object	
genericAnyShape	Tests that the	Test that the	pass
	[smodel/setParams] method	[smodel/setParams] method	
	works with any shape of objects	works for any shape of objects	
	as input.	as input.	
		1) Check that the number of el-	pass
		'chig' 2) Check that each output	
		object contains the correct data	
		2) Robuild the object	
	Tosts that the	Test that the	negg
genericAnyShape	[smodel/setParams] method	[smodel/setParams] method	pass
	works with any shape of objects	works for any shape of objects	
	as input	as input	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	Pass
		'objs' 2) Check that each output	
		object contains the correct data.	
		3) Rebuild the object	
: A C1		· · · ·	pass
genericAnyShape			pass
generic ApyShape			pass
genericAnyonape			pass
genericList	Tests that the	Tests that the	pass
801101101100	[smodel/setParams] method	[smodel/setParams] method	
	works for a list of objects as	works for a list of objects as	
	input.	input.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



## UNIVERSITÀ DEGLI STUDI DI TRENTO

smodel/setParams	5		
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericHistory	Tests that the [smodel/setParams] method properly applies history.	Test that the result of applying the [smodel/setParams] method can be processed back.	pass
		<ol> <li>Check that the last entry in the history of 'out' corre- sponds to '[smodel/setParams]'.</li> <li>Check that the re-built object is the same object as the input.</li> </ol>	pass
genericModify	Tests that the [smodel/setParams] method can modify the input AO.	Test that the [smodel/setParams] method can modify the input object by calling with no output and that the method doesn't change the input of the function notation (with a equal sign).	pass
		1) Check that the modified in- put is changed by the method 2) Check that 'out' and 'obj_eq' are now different. 3) Check that 'obj_eq' is not changed 4) Check that out and amodi are the same	pass
genericOutput	Check that the [smodel/setParams] method pass back the output objects to a list of output variables or to a single variable.	Call the method with a list of output variables and with a sin- gle output variable. Additionaly check that the rebuild method works on the output.	pass
		1) Check that the output con- tains the right number of ob- jects 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass

Table 337: Unit tests for smodel/setParams.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



smodel/setTrans			
minfo	Tests that the getInfo call works for this a general setter method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
genericAnyShape			pass pass
genericAnyShape			pass pass
genericAnyShape			pass
genericAnyShape			pass
genericAnyShape			pass
genericAnvShape			pass pass
genericAnyShape			pass pass
genericAnysnape			pass
genericList	Tests that the [smodel/setTrans] method works for a list of objects as input.	Tests that the [smodel/setTrans] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericList	Tests that the [smodel/setTrans] method works for a list of objects as input.	Tests that the [smodel/setTrans] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericHistory	Tests that the [smodel/setTrans] method properly applies history.	Test that the result of applying the [smodel/setTrans] method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[smodel/setTrans]'. 2) Check that the re-built object is the same object as the input.	pass
genericModify	Tests that the [smodel/setTrans] method can modify the input AO.	Test that the [smodel/setTrans] method can modify the input object by calling with no out- put and that the method doesn't change the input of the function notation (with a equal sign).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



### UNIVERSITÀ DEGLI STUDI DI TRENTO

smodel/setTrans			
		1) Check that the modified in-	pass
		put is changed by the method	
		2) Check that 'out' and 'obj_eq'	
		are now different. 3) Check that	
		'obj_eq' is not changed 4) Check	
		that out and amodi are the same	
man ania Quet nuet	Check that the	Call the method with a list of	pass
genericOutput	[smodel/setTrans] method pass	output variables and with a sin-	
	back the output objects to a list	gle output variable. Additionaly	
	of output variables or to a single	check that the rebuild method	
	variable.	works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 338: Unit tests for smodel/setTrans.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

smodel/setValues			
minfo	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this a general setter method.	for no sets, all sets, and each set	
		1) Check that getInfo call re-	nage
		turned an minfo object in all	pass
		cases 2) Check that all plists	
		have the correct parameters	
		have the correct parameters.	nage
genericAnyShape			pass
genericAnyShape			pass
		Trata that the	pass
genericList	lests that the	lests that the	pass
	[smodel/set values] method	[smodel/set values] method	
	works for a list of objects as	works for a list of objects as	
	input.	1) Charle that the member of all	
		1) Check that the humber of el-	pass
		'chig' 2) Check that each autput	
		objs 2) Check that each output	
	Tests that the	Tests that the	
genericList	Iests that the	Iests that the	pass
	[Smodel/set values] method	[Smodel/set values] method	
	works for a list of objects as	works for a list of objects as	
	input.	1) Check that the number of al	-
		1) Check that the humber of el-	pass
		'chig' 2) Check that each output	
		object contains the correct date	
	Tests that the	Toget that the regult of applying	pagg
genericHistory	[amodel/actValues] method	the [gmodel/getVelves] method	pass
	properly applies history	an he processed had	
	property applies instory.	1) Check that the last entry in	nagg
		the history of 'out' corresponde	pass
		the instory of out corresponds	
		that the re built object is the	
		same object as the input	
	Tests that the	Test that the [smodel/setValues]	nass
genericModify	[smodel/setValues] method can	method can modify the input	Pass
	modify the input AO	object by calling with no out-	
	moung the input ite.	put and that the method doesn't	
		change the input of the function	
		notation (with a equal sign)	
		1) Check that the modified in-	pass
		put is changed by the method	
		2) Check that 'out' and 'obi eq'	
		are now different. 3) Check that	
		'obj_eq' is not changed 4) Check	
		that out and amodi are the same	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

smodel/setValues			
gonoricOutput	Check that the	Call the method with a list of	pass
genericOutput	[smodel/setValues] method pass	output variables and with a sin-	
	back the output objects to a list	gle output variable. Additionaly	
	of output variables or to a single	check that the rebuild method	
	variable.	works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 339: Unit tests for smodel/setValues.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



smodel/setXunits			
minfo	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this a general setter method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
			pass
genericAnyShape			pass
			pass
genericAnySnape			pass
monomic A nur Chama			pass
genericAnyShape			pass
ronoviaAnyChana			pass
genericAnyShape			pass
gonoric Any Shapo			pass
genericAnySnape			pass
gonoric Any Shapo			pass
genericAnyShape			pass
generic AnyShape			pass
generierniyonape			pass
genericList	Tests that the	Tests that the	pass
Seliciterist	[smodel/setXunits] method	[smodel/setXunits] method	
	works for a list of objects as	works for a list of objects as	
	input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'objs' 2) Check that each output	
		object contains the correct data.	
genericList	Tests that the	Tests that the	pass
0	[smodel/setXunits] method	[smodel/setXunits] method	
	works for a list of objects as	works for a list of objects as	
	input.	input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		objs 2) Cneck that each output	
	Tests that the	Test that the regult of applicate	Deca
genericHistory	[amodel/actYunital method	the [amodel/actVunite] method	pass
	properly applies history	can be processed back	
	property applies instory.	1) Check that the last entry in	nase
		the history of 'out' corresponds	pass
		to '[smodel/setXunits]' 2) Check	
		that the re-built object is the	
		same object as the input	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



### UNIVERSITÀ DEGLI STUDI DI TRENTO

smodel/setXunits			
mon onio Mo dife	Tests that the	Test that the [smodel/setXunits]	pass
genericitiodity	[smodel/setXunits] method can	method can modify the input	
	modify the input AO.	object by calling with no out-	
		put and that the method doesn't	
		change the input of the function	
		notation (with a equal sign).	
		1) Check that the modified in-	pass
		put is changed by the method	
		2) Check that 'out' and 'obj_eq'	
		are now different. 3) Check that	
		'obj_eq' is not changed 4) Check	
		that out and amodi are the same	
mon onic Output	Check that the	Call the method with a list of	pass
genericOutput	[smodel/setXunits] method pass	output variables and with a sin-	
	back the output objects to a list	gle output variable. Additionaly	
	of output variables or to a single	check that the rebuild method	
	variable.	works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 340: Unit tests for smodel/setXunits.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



smodel/setXvals			
minfo	Tests that the getInfo call works for this a general setter method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
genericAnyShape			pass pass
genericAnyShape			pass pass
genericAnyShape			pass
genericAnyShape			pass
genericAnyShape			pass pass
genericApyShape			pass pass
genericAnyshape			pass
genericAnyShape			pass
genericList	Tests that the [smodel/setXvals] method works for a list of objects as input.	Tests that the [smodel/setXvals] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericList	Tests that the [smodel/setXvals] method works for a list of objects as input.	Tests that the [smodel/setXvals] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericHistory	Tests that the [smodel/setXvals] method properly applies history.	Test that the result of applying the [smodel/setXvals] method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[smodel/setXvals]'. 2) Check that the re-built object is the same object as the input.	pass
genericModify	Tests that the [smodel/setXvals] method can modify the input AO.	Test that the [smodel/setXvals] method can modify the input object by calling with no out- put and that the method doesn't change the input of the function notation (with a equal sign).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



## UNIVERSITÀ DEGLI STUDI DI TRENTO

smodel/setXvals			
		1) Check that the modified in-	pass
		put is changed by the method	
		2) Check that 'out' and 'obj_eq'	
		are now different. 3) Check that	
		'obj_eq' is not changed 4) Check	
		that out and amodi are the same	
man ani a Quet nuet	Check that the	Call the method with a list of	pass
genericOutput	[smodel/setXvals] method pass	output variables and with a sin-	
	back the output objects to a list	gle output variable. Additionaly	
	of output variables or to a single	check that the rebuild method	
	variable.	works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 341: Unit tests for smodel/setXvals.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



smodel/setXvar			
minfo	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this a general setter method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
generic ApyShape			pass
generierinyonape			pass
generic AnyShape			pass
generici inyonape			pass
generic AnyShape			pass
generierinybhape			pass
generic AnyShape			pass
generierinyonape			pass
generic Any Shape			pass
generierinyonape			pass
generic AnyShape			pass
generierinyonape			pass
generic Any Shape			pass
generierinyonape			pass
gonorieL ist	Tests that the [smodel/setXvar]	Tests that the [smodel/setXvar]	pass
genericList	method works for a list of	method works for a list of objects	
	objects as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'objs' 2) Check that each output	
		object contains the correct data.	
genericList	Tests that the [smodel/setXvar]	Tests that the [smodel/setXvar]	pass
generichist	method works for a list of	method works for a list of objects	
	objects as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'objs' 2) Check that each output	
		object contains the correct data.	
genericHistory	Tests that the [smodel/setXvar]	Test that the result of apply-	pass
Senerioring	method properly applies history.	ing the [smodel/setXvar] method	
		can be processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[smodel/setXvar]'. 2) Check	
		that the re-built object is the	
		same object as the input.	
genericModify	[ Iests that the [smodel/setXvar]	[ lest that the [smodel/setXvar]	pass
	method can modify the input	method can modify the input	
	AO.	object by calling with no out-	
		put and that the method doesn't	
		change the input of the function	
		notation (with a equal sign).	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



## UNIVERSITÀ DEGLI STUDI DI TRENTO

smodel/setXvar			
		1) Check that the modified in-	pass
		put is changed by the method	
		2) Check that 'out' and 'obj_eq'	
		are now different. 3) Check that	
		'obj_eq' is not changed 4) Check	
		that out and amodi are the same	
mon oni o Outmut	Check that the [smodel/setXvar]	Call the method with a list of	pass
genericOutput	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 342: Unit tests for smodel/setXvar.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



smodel/setYunits			
minfo	Tests that the getInfo call works for this a general setter method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
genericAnyShape			pass pass
genericAnyShape			pass
genericList	Tests that the [smodel/setYunits] method works for a list of objects as input.	Tests that the [smodel/setYunits] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericList	Tests that the [smodel/setYunits] method works for a list of objects as input.	Tests that the [smodel/setYunits] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'objs' 2) Check that each output object contains the correct data.	pass
genericHistory	Tests that the [smodel/setYunits] method properly applies history.	Test that the result of applying the [smodel/setYunits] method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[smodel/setYunits]'. 2) Check that the re-built object is the same object as the input.	pass
genericModify	Tests that the [smodel/setYunits] method can modify the input AO.	Test that the [smodel/setYunits] method can modify the input object by calling with no out- put and that the method doesn't change the input of the function notation (with a equal sign).	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



### UNIVERSITÀ DEGLI STUDI DI TRENTO

smodel/setYunits			
		1) Check that the modified in-	pass
		put is changed by the method	
		2) Check that 'out' and 'obj_eq'	
		are now different. 3) Check that	
		'obj_eq' is not changed 4) Check	
		that out and amodi are the same	
man ania Quet nuet	Check that the	Call the method with a list of	pass
genericOutput	[smodel/setYunits] method pass	output variables and with a sin-	
	back the output objects to a list	gle output variable. Additionaly	
	of output variables or to a single	check that the rebuild method	
	variable.	works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 343: Unit tests for smodel/setYunits.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



smodel/smodel			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.0	Tests that the [smodel/smodel]	Test that the [smodel/smodel]	pass
02	method works with a vector of	method works for a vector of ob-	
	objects as input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	1
		'vec' 2) Check that each output	
		object contains the correct data.	
	Tests that the [smodel/smodel]	Test that the [smodel/smodel]	pass
03	method works with a matrix of	method works for a matrix of ob-	1
	objects as input.	jects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	1
		'mat' 2) Check that each output	
		object contains the correct data.	
	Tests that the [smodel/smodel]	Test that the [smodel/smodel]	pass
04	method works with a list of	method works for a list of objects	1
	objects as input.	as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	-
		'mat' 2) Check that each output	
		object contains the correct data.	
	Tests that the [smodel/smodel]	Tests that the [smodel/smodel]	pass
05	method works with a mix of	method works with a mix of dif-	1
	different arrays of objects as	ferent arrays of objects as input.	
	input.	1) Check that the number of el-	pass
		ements in 'out' is the same as in	1
		'mat' 2) Check that each output	
		object contains the correct data.	
	Tests that the [smodel/smodel]	Test that the result of applying	pass
06	method properly applies history.	the [smodel/smodel] method can	1
		be processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	1
		to '[smodel]/smodel]'. 2) Check	
		that the re-built object is the	
		same object as the input.	
1	1	i a sultant and the part.	1



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



$\rm smodel/smodel$			
07	Tests that the smodel can read older MAT files which have different values in 'xvals'	Tests that the smodel can read older MAT files which have different values in 'xvals' The smodel object was created with the com- mands: $s = smodel('A.*(t >= toff)'); s.setName('A.*(tf', 5,300);s.setParams('A','toff', 5,300);s.setParams('A','toff', 5,300);s.setDescription('Step at toff');s.setDescription('Step func-tion of amplitude A at timetoff'); s.setXunits('Hz1/2km2'); s.setXunits('Hz1/2km2'); s.setXunits('iz2m^1'); s.setXvals(1:1e3);s.setAliasNames('a');s.setAliasValues(8);$	pass
		1) Check that the objects are quite the same 2) Check that 'xvals' is a cell-array with the numbers from 1 to 1000	pass
60	Tests that the constructor method doesn't apply history to the read MAT-file constructor.	Tests that the constructor method doesn't apply history to the read MAT-file constructor. 1) Check that the history is the same as the history of the saved object. Because save and load shouldn't add a history step. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
61	Tests that the constructor properly applies history to the read XML-file constructor.	Tests that the constructor prop- erly applies history to the read XML-file constructor. 1) Check that the history is the same as the history of the saved object. Because save and load shouldn't add a history step. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
62	Tests that the constructor properly applies history in the struct constructor.	<ul> <li>Tests that the constructor properly applies history in the struct constructor.</li> <li>1) Check that the last entry in the history of 'out' corresponds to the class name. 2) Check that the 'rebuild' method produces the same object as 'out'.</li> </ul>	pass
64	Tests that the contructor properly applies history to the plist(filename) constructor.	Tests that the contructor prop- erly applies history to the plist(filename) constructor.	pass


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

smodel/smodel			
		1) Check that the save method	pass
		doesn't change the input ob-	
		ject 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
65	Tests that the contructed object	Tests that the contructed object	pass
05	can be submitted and retrieved.	can be submitted and retrieved.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to the class name. 2) Check that	
		the 'rebuild' method produces	
		the same object as 'out'.	
69	Tests that the constructor	Tests that the constructor	pass
00	properly applies history to the	properly applies history to the	
	conn+Id constructor.	conn+Id constructor.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to class name. 2) Check that the	
		'rebuild' method produces the	
		same object as 'out'.	

Table 344: Unit tests for smodel/smodel.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



#### UNIVERSITÀ DEGLI STUDI DI TRENTO

ssm/addParamete	ers		
02	Tests that the [ssm/addParameters] method works with a vector of objects as input.	Testthatthe[ssm/addParameters]methodworks for a vector of objects asinput.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'vec' 2) Check that each output object contains the correct data.	pass
03	Tests that the [ssm/addParameters] method works with a matrix of objects as input.	Testthatthe[ssm/addParameters]methodworks for a matrix of objects asinput.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
04	Tests that the [ssm/addParameters] method works with a list of objects as input.	Test that the [ssm/addParameters] method works for a list of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
04	Tests that the [ssm/addParameters] method works with a list of objects as input.	Testthatthe[ssm/addParameters]methodworks for a list of objects asinput.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
05	Tests that the [ssm/addParameters] method works with a mix of different arrays of objects as input.	Teststhatthe[ssm/addParameters]methodworkswith a mix of differentarrays of objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'mat' 2) Check that each output object contains the correct data.	pass
06	Tests that the [ssm/addParameters] method properly applies history.	Test that the result of applying the [ssm/addParameters] method can be processed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to '[ssm/addParameters]'. 2) Check that the re-built object is the same object as the input.	pass

Table 345: Unit tests for  $\rm ssm/addParameters.$ 



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

ssm/copy			
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	

Table 346: Unit tests for ssm/copy.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



ssm/loadobj			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check the shape of the loaded	pass
		objects.	

Table 347: Unit tests for ssm/loadobj.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

model/DFACS			
001	Tests that the model	Test that the model <model></model>	pass
901	<model> has a meaningful</model>	has a meaningful name.	
	name.	1) Check that the name of the	pass
		object is not empty and not equal	
		to 'none'	
002	Tests that the model	Test that the model <model></model>	pass
902	<model> has a meaningful</model>	has a meaningful description.	
	description.	1) Check that the description of	pass
		the object is not empty and not	
		equal to 'none'	
003	Tests that the model	Test that the model <model></model>	pass
900	<model> responds to 'DIM'</model>	responds to 'DIM' configuration	
	configuration key.	key.	
		1) Check that the model builds	pass
		with DIM==1 2) Check that the	
		model builds with $DIM==2$ 3)	
		Check that the model builds with	
		DIM= $=3$ 4) Check that the 3	
		models are different	

Table 348: Unit tests for model/DFACS.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

model/IFO			
001	Tests that the model	Test that the model <model></model>	pass
901	<model> has a meaningful</model>	has a meaningful name.	
	name.	1) Check that the name of the	pass
		object is not empty and not equal	
		to 'none'	
002	Tests that the model	Test that the model <model></model>	pass
902	<model> has a meaningful</model>	has a meaningful description.	
	description.	1) Check that the description of	pass
		the object is not empty and not	
		equal to 'none'	
002	Tests that the model	Test that the model <model></model>	pass
905	<model> responds to 'DIM'</model>	responds to 'DIM' configuration	
	configuration key.	key.	
		1) Check that the model builds	pass
		with DIM==1 2) Check that the	
		model builds with $DIM==2$ 3)	
		Check that the model builds with	
		DIM= $=3$ 4) Check that the 3	
		models are different	

Table 349: Unit tests for model/IFO.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



## UNIVERSITÀ DEGLI STUDI DI TRENTO

ssm/ssm			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
	Tests that the [ssm/ssm]	Test that the [ssm/ssm] method	pass
02	method works with a vector of	works for a vector of objects as	
	objects as input.	input.	
	T J T T	1) Check that the number of el-	pass
		ements in 'out' is the same as in	Puss
		'vec' 2) Check that each output	
		object contains the correct data	
	Tests that the [sem/sem]	Test that the [ssm/ssm] method	nass
03	mothod works with a matrix of	works for a matrix of objects as	pass
	objects as input	input	
	objects as input.	1) Check that the number of el-	nase
		ements in 'out' is the same as in	pass
		'mat' 2) Check that each output	
		object contains the correct data	
	Tests that the [gam /gam]	Test that the [ggm/ggm] method	negg
04	method monks with a list of	rest that the [ssin/ssin] method	pass
	shipping ag input	works for a list of objects as in-	
	objects as input.	put.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		mat ² ) Check that each output	
		object contains the correct data.	
05	Tests that the [ssm/ssm]	Tests that the [ssm/ssm] method	pass
	method works with a mix of	works with a mix of different ar-	
	different arrays of objects as	rays of objects as input.	
	input.	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		(mat' 2) Check that each output	
		object contains the correct data.	
06	Tests that the [ssm/ssm]	Test that the result of apply-	pass
	method properly applies history.	ing the [ssm/ssm] method can be	
		processed back.	
		1) Check that the last entry in	pass
		the history of 'out' corresponds	
		to '[ssm/ssm]'. 2) Check that the	
		re-built object is the same object	
		as the input.	
60	Tests that the constructor	Tests that the constructor	pass
	method doesn't apply history to	method doesn't apply history to	
	the read MAT-file constructor.	the read MAT-file constructor.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



#### UNIVERSITÀ DEGLI STUDI DI TRENTO

$\rm ssm/ssm$			
		1) Check that the history is the same as the history of the saved object. Because save and load shouldn't add a history step. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
61	Tests that the constructor properly applies history to the read XML-file constructor.	Tests that the constructor prop- erly applies history to the read XML-file constructor. 1) Check that the history is the same as the history of the saved object. Because save and load shouldn't add a history step. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
62	Tests that the constructor properly applies history in the struct constructor.	<ul> <li>Tests that the constructor properly applies history in the struct constructor.</li> <li>1) Check that the last entry in the history of 'out' corresponds to the class name. 2) Check that the 'rebuild' method produces the same object as 'out'.</li> </ul>	pass
63	Tests that the constructor properly applies history to the pzmodel constructor.	<ul> <li>Tests that the constructor properly applies history to the pz-model constructor.</li> <li>1) Check that the last entry in the history of 'out' corresponds to the class name. 2) Check that the 'rebuild' method produces the same object as 'out'.</li> </ul>	pass
64	Tests that the contructor properly applies history to the plist(filename) constructor.	Tests that the contructor prop- erly applies history to the plist(filename) constructor. 1) Check that the save method doesn't change the input ob- ject 2) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
65	Tests that the contructed object can be submitted and retrieved.	Tests that the contructed object can be submitted and retrieved. 1) Check that the last entry in the history of 'out' corresponds to the class name. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
66	Tests that the constructor properly works with the plist(pzmodel) constructor.	Tests that the constructor properly works with the plist(pzmodel) constructor.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



# università degli studi di trento

ssm/ssm			
		1) Check that the last entry in the history of 'out' corresponds to 'ssm'. 2) Check that the 're- build' method produces the same object as 'out'.	pass
67	Tests that the constructor properly applies history to the pole/zero model + plist object constructor.	Tests that the constructor properly applies history to the pole/zero model + plist object constructor.	pass
		1) Check that the last entry in the history of 'out' corresponds to class name. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
68	Tests that the constructor properly applies history to the conn+Id constructor.	Tests that the constructor properly applies history to the conn+Id constructor.	pass
		1) Check that the last entry in the history of 'out' corresponds to class name. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass

Table 350: Unit tests for ssm/ssm.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



## UNIVERSITÀ DEGLI STUDI DI TRENTO

time/datenum			
901	Tests the 'datenum' method.	Create a time object and use datenum to produce a serial date number.	pass
		Check that the returned value is	pass
		using time/format with the same	
		format string when converted us-	
		ing MATLAB's datestr() func-	
		tion.	
901	Tests the 'datenum' method	Create a time object and use	pass
		datenum to produce a serial date	
		number.	
		Check that the returned value is	pass
		the same as the one you get when	
		using time/format with the same	
		format string when converted us-	
		ing MATLAB's datestr() func-	
		tion.	
903	Tests 'double' method	Set the time-zone to PST and	pass
500	Tests double method.	create a time object and use	
		datenum to produce a serial date	
		number.	
		Check that the returned value is	pass
		the same as the one you get when	
		using time/format with the same	
		format string when converted us-	
		ing MATLAB's datestr() func-	
		tion.	

Table 351: Unit tests for time/datenum.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



time/double			
901	Tests 'double' method	Use the double() method on sin-	pass
001	Tests double method.	gle time objects and vectors of	
		time objects.	
		Check that double returns the	pass
		expected numerical values for	
		each case.	

Table 352: Unit tests for time/double.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



time/format			
001	Tests 'format' static mathed	Use the static format method to	pass
501	rests format static method.	produce a number of time strings	
		in different formats and for dif-	
		ferent time-zones.	
		Check that the returned strings	pass
		match the expected strings.	

Table 353: Unit tests for time/format.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



time/minus			
01	Tests time object minus	Compute the difference between	pass
	operator.	time objects and doubles.	
		Check the resulting time objects	pass
		have the correct values.	

Table 354: Unit tests for time/minus.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



#### UNIVERSITÀ DEGLI STUDI DI TRENTO

time/parse			
001	Tests 'parse' static method with	Use parse() to parse different	pass
501	time string only.	time strings with different time-	
		zones set.	
		Check the resulting millsecond	pass
		values are the expected ones.	
002	Tests 'parse' static method with	Use parse() to parse different	pass
502	time and format strings.	time strings with different time-	
		zones set. The time-string format	
		is specified as a second input ar-	
		gument.	
		Check the resulting millsecond	pass
		values are the expected ones.	
002	Tests 'parse' static method with	Use parse() to parse different	pass
900	time string and numeric format.	time strings using the supported	
		MATLAB numeric time formats.	
		Check the resulting millsecond	pass
		values against the result of pars-	
		ing a time string. (See time/parse	
		utp_902.)	
004	Tests 'parse' static method with	Use parse() to parse different	pass
904	time string containing timezone	time strings which contain the	
	information.	time-zone.	
		Check the resulting millisecond	pass
		value is the expected one.	
005	Tests 'parse' static method with	Use parse() to parse different	pass
300	time string and timezone	time strings when passing the	
	specification.	time-zone as the third argument.	
		The second argument (the for-	
		mat) is left empty.	
		Check the resulting millisecond	pass
		value is the expected one.	

Table 355: Unit tests for time/parse.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



time/plus			
101	Tests time object plus operator.	Compute the sum of time objects and doubles.	pass
		Check the resulting time objects have the correct values.	pass

Table 356: Unit tests for time/plus.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



time/string			
001	Tests that the output of the	Use string to convert a time ob-	pass
901	'string' method can be used to	ject to a string. Use eval on the	
	recreate the time object.	result to recreate a time object.	
		Check the recreated time object	pass
		matches the original.	

Table 357: Unit tests for time/string.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



## UNIVERSITÀ DEGLI STUDI DI TRENTO

time/time			
901	Tests time object constructor	Call the time() constructor with	pass
501	without arguments. Should	no inputs.	
	return the current time.	No test can be done here since	pass
		we don't know the time when the	
		constructor is called. We could	
		check that the resulting time is	
		> 0, but that's not so useful.	
902	Tests time object constructor	Call the time() constructor with	pass
502	from numeric value.	a numeric input (number of sec-	
		onds since epoch).	
		Check the time object has the ex-	pass
		pected millisecond value.	
903	Tests time object constructor	Call the time() constructor with	pass
500	from string.	different time-string inputs.	
		Check the time objects have the	pass
		expected millisecond values.	
904	Tests time object constructor	Call the time() constructor with	pass
001	from a cell array of strings.	cell-array of time-strings.	
		Check the time objects have the	pass
		expected millisecond values.	
905	Tests time object constructor	Call the time() constructor with	pass
	from plist with 'time'	plist input. Check a plist with a	
	parameter.	numeric value and a plist with a	
		time-string.	
		Check the time objects have the	pass
		expected millisecond values.	
906	Tests time object constructor	Call the time() constructor with	pass
	from plist with 'milliseconds'	plist input using the 'millisecond'	
	parameter	key.	<u> </u>
		Check the time object has the ex-	pass
		pected millisecond value.	
907	Tests time object constructor	Call the time() constructor with	pass
	from plist with more	plist input containing the time	
	parameters.	String and the time-zone. key.	
		Check the time objects have the	pass
	Testa time object constructor	Call the time() constructor with	
908	from structure	call the time() constructor with	pass
	from structure.	alling structure obtained from	
		kov	
		Check the original and recon	nagg
		structed time objects have the	pass
		same millisecond value	
	Tests time object constructor	Call the time() constructor with	nase
909	from string and time format	an input time string and time	pass
	nom sumg and unle lormat.	format string	
		Check the resulting time of	nase
		ject has the expected millisecond	Pass
		value	
		varao.	1



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



time/time

Table 358: Unit tests for time/time.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



### UNIVERSITÀ DEGLI STUDI DI TRENTO

time/timeformat			
001	Tests that the method really	Check that the timeformat	pass
901	returns what is in the user	method runs without error.	
	preferences.	Check that the returned value is	pass
		the same as the one set in the	
		user preferences.	
002	Change user preferences and do	Set the time string format in the	pass
902	it again.	preferences and check that the	
		timeformat method runs without	
		error.	
		Check that the returned value is	pass
		the same as the one set in the	
		user preferences.	
002	Change user preferences and do	Set the time string format in the	pass
905	it again.	preferences and check that the	
		timeformat method runs without	
		error.	
		Check that the returned value is	pass
		the same as the one set in the	
		user preferences.	

Table 359: Unit tests for time/timeformat.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



## UNIVERSITÀ DEGLI STUDI DI TRENTO

time/timezone			
001	Tests that the method really	Call the static timezone method.	pass
901	returns what is in the user	Check the resulting timezone ob-	pass
	preferences.	ject against the one in the user	
		preferences.	
002	Change user preferences and do	Set the timezone in the prefer-	pass
902	it again.	ences then call the static time-	
		zone method.	
		Check the resulting timezone ob-	pass
		ject against the one in the user	
		preferences.	
002	Change user preferences and do	Set the timezone in the prefer-	pass
905	it again.	ences then call the static time-	
		zone method.	
		Check the resulting timezone ob-	pass
		ject against the one in the user	
		preferences.	

Table 360: Unit tests for time/timezone.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

timespan/char			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the char method	Test that the char method works	pass
02	works with a vector of	for a vector of TIMESPAN ob-	
	TIMESPAN objects as input.	jects as input.	
		1) Check that the output contain	pass
		at least each start time	
03	Tests that the char method	Test that the char method works	pass
00	works with a matrix of	for a matrix of TIMESPAN ob-	
	TIMESPAN objects as input.	jects as input.	
		1) Check that the output contain	pass
		at least each start time	
04	Tests that the char method	Test that the char method works	pass
01	works with a list of TIMESPAN	for a list of TIMESPAN objects	
	objects as input.	as input.	
		1) Check that the output contain	pass
		at least each start time	
05	Tests that the char method	Test that the char method works	pass
	works with a mix of different	with an input of matrices and	
	shaped TIMESPAN objects as	vectors and single TIMESPAN	
	input.	objects.	
		1) Check that the output contain	pass
		at least each start time	
06	Tests that the char method	The method char doesn't change	pass
	properly applies history.	the data, thus it is not possible	
		to check the history. Nothing to	
		do.	
			pass

Table 361: Unit tests for timespan/char.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

timespan/copy			
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	
60	Tests that the copy method	Test the positive (copy-case)	pass
09	works with a single object as an	and the negative (non copy-case)	
	input.	case.	
		1) Check that the output is a	pass
		'real' copy or only a copy of the	
		handle	

Table 362: Unit tests for timespan/copy.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

timespan/created			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters	pass
02	Tests that the created method works with a vector of TIMESPAN objects as input.	Test that the created method works for a vector of TIMESPAN objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'tsv' 2) Check that each output contains the correct data.	pass
03	Tests that the created method works with a matrix of TIMESPAN objects as input.	Test that the created method works for a matrix of TIMES- PAN objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'tsm' 2) Check that each output contains the correct data.	pass
04	Tests that the created method works with a list of TIMESPAN objects as input.	Test that the created method works for a list of TIMESPAN objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the created method works with a mix of different shaped TIMESPAN objects as input.	Test that the created method works with an input of matrices and vectors and single TIMES- PAN objects.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the created method properly applies history	This method doesn't change the input object, thus no history is added to the object.	pass
07	Tests that the created method can be used with the modify command.	Tests that the created method can be used with the modify command. 1) Check the single object 2)	pass pass pass
08	Tests that the created method retruns always a well defined time object even for an empty input object.	Check the matrix object Test that the created method with an empty 'TIMESPAN ob- ject	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



timespan/created		
	1) Check that the output is a	pass
	time object with a ell defined	
	time.	

Table 363: Unit tests for timespan/created.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

timespan/creator			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.0	Tests that the creator method	Test that the creator method	pass
02	works with a vector of	works for a vector of TIMESPAN	-
	TIMESPAN objects as input.	objects as input.	
	0 1	1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'tsy' 2) Check that each output	
		contains the correct data.	
	Tests that the creator method	Test that the creator method	nass
03	works with a matrix of	works for a matrix of TIMES-	Pass
	TIMESPAN objects as input	PAN objects as input	
		1) Check that the number of el-	nass
		ements in 'out' is the same as in	pass
		'tsm' 2) Check that each output	
		contains the correct data	
	Tests that the creater method	The creater method doesn't work	nnee
04	works with a list of TIMESDAN	for a list of TIMESPAN abjects	pass
	objects as input	as input	
	objects as input.	1) Nothing to tost	neg
	Tests that the exector method	The greater method description	pass
05	rests that the creator method	with different shaned input sh	pass
	shaped TIMESDAN shipets as	isota	
	snaped TIMESPAN objects as	Jects.	
		1) Nothing to test	pass
06	lests that the creator method	I his method doesn't change the	pass
	properly applies history	input object, thus no history is	
		added to the object.	
		1) Nothing to check.	pass
07	Tests that the creator method	Tests that the creator method	pass
	can be used with the modify	can be used with the modify	
	command.	command.	
		1) Check the single object 2)	pass
		Check the matrix object	
08	Tests that the creator method	Test that the creator method	pass
	retruns all	uses the option 'all' direct or in	
	creator(s)/modifier(s) which are	a plist. The test file must have	
	in the history.	the modifier 'first', 'second' and	
		'third'	
		1) Check that out1 contains only	pass
		one creator 2) Check that out2	
		contain more creator/modifier	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



timespan/creator			
09	Tests the negative case for the option 'all'.	Test that the creator method throws an error if the option 'all' is used in connection with a ma- trix/vector of TIMESPAN ob- jects.	pass
		1) Nothing to test.	pass

Table 364: Unit tests for timespan/creator.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

timespan/display			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the display method	Test that the display method	pass
02	works with a vector of	works for a vector of TIMESPAN	
	TIMESPAN objects as input.	objects as input.	
		1) Check that the output contain	pass
		at least each object name	
03	Tests that the display method	Test that the display method	pass
00	works with a matrix of	works for a matrix of TIMES-	
	TIMESPAN objects as input.	PAN objects as input.	
		1) Check that the output contain	pass
		at least each object name	
04	Tests that the display method	Test that the display method	pass
	works with a list of TIMESPAN	works for a list of TIMESPAN	
	objects as input.	objects as input.	
		1) Check that the output contain	pass
		at least each object name	
05	Tests that the display method	Test that the display method	pass
	works with a mix of different	works with an input of matrices	
	shaped TIMESPAN objects as	and vectors and single TIMES-	
	input.	PAN objects as.	
		1) Check that the output contain	pass
		at least each object name	
06	Tests that the display method	The method display doesn't	pass
00	properly applies history.	change the data, thus it is not	
		possible to check the history.	
		Nothing to do.	
			pass

Table 365: Unit tests for timespan/display.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

timespan/eq			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the eq method works with a vector of TIMESPAN objects as input.	Test that the eq method works for a vector of TIMESPAN ob- jects as input. Test the positive and the negative case.	pass
	Tests that the eq method works	function.	pass
03	with a matrix of TIMESPAN objects as input.	for a matrix of TIMESPAN objects as input. Test the positive and the negative case.	
		1) Check the output of the eq function.	pass
04	Tests that the eq method works with a list of TIMESPAN objects as input.	The eq method doesn't works for a list of TIMESPAN objects as input. Nothing to do.	pass
			pass
05	Tests that the eq method works with a mix of different shaped TIMESPAN objects as input.	The eq method doesn't works for a list of TIMESPAN objects as input. Nothing to do.	pass
			pass
06	Tests that the eq method properly applies history.	The eq method doesn't change the TIMESPAN object, thus will no history added. Nothing to do	pass
			pass
07	Test the eq method with an exception list. With the LTPDA toolbox 2.0 it is only possible to test the exception list with properties where a public set method exist.	Test the eq method with the exception 'name'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because ts is created at an other time. 1) Check the output.	pass
	Test the eq method with an	Test the eq method with the ex-	pass
08	exception list. With the LTPDA toolbox 2.0 it is only possible to test the exception list with properties where a public set method exist.	ception 'endt'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because ts is created at an other time. 1) Check the output.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

timespan/eq			
00	Test the eq method with an	Test the eq method with the ex-	pass
09	exception list. With the LTPDA	ception 'startT'. Use the option	
	toolbox 2.0 it is only possible to	'internal' to suppress the history.	
	test the exception list with	It is necessary to add 'created'	
	properties where a public set	to the exception list because ts	
	method exist.	is created at an other time.	
		1) Check the output.	pass
10	Test the eq method with an	Test that the eq method uses the	pass
12	exception list which is in a plist.	exception list in a plist.	
		1) Check the output.	pass

Table 366: Unit tests for timespan/eq.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

timespan/get			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests the get method of the	Test that the get returns returns	pass
02	timespan class.	the value of the specified prop-	
		erty. Do this for all properties of	
		the TIMESPAN object.	
		1) Check the correct value of the	pass
		output	
03	Tests that the get method works	Test that the get returns returns	pass
05	with a plist.	the value of the specified prop-	
		erty which is defined in a plist.	
		1) Check the correct value of the	pass
		output	
04	Tests the get method of the	Test that the get throws an error	pass
04	timespan class.	if the input are more than one	
		TIMESPAN object.	
		1) Nothing to test	pass

Table 367: Unit tests for timespan/get.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

timespan/index			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set	pass
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the index method	Test that the index method	pass
	works with a vector of	works for a vector of TIMESPAN	
	TIMESPAN objects as input.	objects as input. The following	
		indexing should work: $I = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$	
		$\int \text{or } (1/J) = [(1,1), (1,2), (1,3)]$	
		1) Check that the index method	pass
		selects the correct object.	<u> </u>
03	Tests that the index method	Test that the index method	pass
	works with a matrix of	works for a matrix of TIMES-	
	TIMESPAN objects as input.	PAN objects as input. The fol-	
		lowing indexing should work: $I = \begin{bmatrix} 1 & 2 & 5 \end{bmatrix}$ and $\begin{bmatrix} I & I \end{bmatrix}$	
		$\begin{bmatrix} 1 & 3 & 5 \end{bmatrix}$ or $(1/J) = \begin{bmatrix} (1,1), (1,2), \\ (1,2) \end{bmatrix} \begin{bmatrix} 2 & 4 & 6 \end{bmatrix} \begin{bmatrix} (2,1), (2,2), (2,2) \end{bmatrix}$	
		$\begin{array}{ c c c c c c c c c c c c c c c c c c c$	-
		1) Check that the index method	pass
	Trata that the index weathed	The index wethed do en 't ment	
04	rests that the index method	for a list of TIMESDAN abjects	pass
	works with a list of TIMESFAN	ag input	
	objects as input.	1) Nothing to tost	nase
	Tests that the index method	Tost that the result of index have	pass
05	properly applies history	an additional history step	pass
	property applies instory.	1) Check that the last entry in	nass
		the history of 'out' corresponds	pass
		to 'index'	
	Tests that the index method	Tests that the index method	pass
06	works for the modifier	works for the modifier command.	pass
	command.	1) Check that the history-plist	pass
		contains the used indices. 2)	P
		Check that the index method se-	
		lects the correct object	
07		Test that the index method can	pass
07	Control the method with a plist.	be controled with a plist.	
		1) Check that the history-plist	pass
		contains the used indices. 2)	
		Check that the index method se-	
		lects the correct object	
08	Test that the index method	Test that the index method se-	pass
00	selects more objects if I have	lects more objects if I have more	
	more indices.	indices.	
		1) Check that the history-plist	pass
		contains the used indices. 2)	
		Check that the index method se-	
		lects the correct object	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



timespan/index

Table 368: Unit tests for timespan/index.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

timespan/isprop			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters	pass
02	Tests that the isprop method works with a vector of TIMESPAN objects as input.	Test that the isprop method works for a vector of TIMESPAN objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'tsv' 2) Check that each output contains the correct data.	pass
03	Tests that the isprop method works with a matrix of TIMESPAN objects as input.	Test that the isprop method works for a matrix of TIMES- PAN objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in 'tsm' 2) Check that each output contains the correct data.	pass
04	Tests that the isprop method works with a list of TIMESPAN objects as input.	Test that the isprop method works for a list of TIMESPAN objects as input.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
05	Tests that the isprop method works with a mix of different shaped TIMESPAN objects as input.	Test that the isprop method works with an input of matrices and vectors and single TIMES- PAN objects.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output contains the correct data.	pass
06	Tests that the isprop method properly applies history.	The method isprop doesn't change the object, thus it is not necessary to apply history.	pass
07	Tests that the isprop method works for each property.	Test that the isprop method works for the properties: startT', 'endT', 'timeformat', 'timezone', 'interval' 'hist' and 'name'	pass
		1) Check that each output con- tains the correct data.	pass
08	Test the negative case and the not function command.	Test that the isprop method retrun false for a unknown prop- erty and for methods of the ob- ject.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



timespan/isprop		
	1) Check that each output tains the correct data.	con- pass

Table 369: Unit tests for timespan/isprop.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



timespan/loadobj			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check the shape of the loaded	pass
		objects.	

Table 370: Unit tests for timespan/loadobj.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

timespan/ne			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the ne method works with a vector of TIMESPAN objects as input.	Test that the ne method works for a vector of TIMESPAN ob- jects as input. Test the positive and the negative case.	pass
		1) Check the output of the ne function.	pass
03	Tests that the ne method works with a matrix of TIMESPAN objects as input.	Test that the ne method works for a matrix of TIMESPAN ob- jects as input. Test the positive and the negative case.	pass
		1) Check the output of the ne function.	pass
04	Tests that the ne method works with a list of TIMESPAN objects as input.	The ne method doesn't works for a list of TIMESPAN objects as input. Nothing to do.	pass
05	Tests that the ne method works with a mix of different shaped TIMESPAN objects as input.	The ne method doesn't works for a list of TIMESPAN objects as input. Nothing to do.	pass pass
06	Tests that the ne method properly applies history.	The ne method doesn't change the TIMESPAN object, thus will no history added. Nothing to do	pass pass
07	Test the ne method with an exception list. The function timespan/ne use the function timespan/eq so it is not necessary to check all possibilities of the exception list.	Test the ne method with the exception 'name'. Use the option 'internal' to suppress the history. It is necessary to add 'created' to the exception list because ts is created at an other time. 1) Check that each output con-	pass
08	Test the ne method with an exception list which is in a plist.	tains the correct data.Test that the ne method uses the exception list in a plist.	pass
		1) Check that each output con- tains the correct data.	pass

Table 371: Unit tests for timespan/ne.


Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

timespan/rebuild			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the rebuild method	Test that the rebuild method	pass
02	works with a vector of	works for a vector of TIMESPAN	
	TIMESPAN objects as input.	objects as input.	
		1) Check the rebuilt output.	pass
0.2	Tests that the rebuild method	Test that the rebuild method	pass
05	works with a matrix of	works for a matrix of TIMES-	
	TIMESPAN objects as input.	PAN objects as input.	
		1) Check the rebuilt output.	pass
0.4	Tests that the rebuild method	Test that the rebuild method	pass
04	works with a list of TIMESPAN	works for a list of TIMESPAN	
	objects as input.	objects as input.	
		1) Check the rebuilt output.	pass
05	Tests that the rebuild method	Test that the rebuild method	pass
00	works with a mix of different	works with an input of matrices	
	shaped TIMESPAN objects as	and vectors and single TIMES-	
	input.	PAN objects.	
		1) Check the rebuilt output.	pass
06	Tests that the rebuild method	The method rebuild doesn't	pass
	properly applies history.	change the data, thus it is not	
		possible to check the history.	
		Nothing to do.	
			pass
07	Check that the rebuild method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 372: Unit tests for timespan/rebuild.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

timespan/save			
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the save method works with a vector of TIMESPAN objects as input.	Test that the save method works for a vector of TIMESPAN ob- jects as input. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out1' and 'out2' are the same as in 'tsv' 2) Check that the loaded objects are the same as the saved objects. 3) The out- puts 'out1' and 'out2' must be the same.	pass
03	Tests that the save method works with a matrix of TIMESPAN objects as input.	Test that the save method works for a matrix of TIMESPAN ob- jects as input. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of ele- ments in 'out1' and 'out2' are the same as in 'tsm' 2) Check that the loaded objects are the same as the saved objects. 3) The out- puts 'out1' and 'out2' must be the same.	pass
04	Tests that the save method works with a list of TIMESPAN objects as input.	Test that the save method works for a list of TIMESPAN objects as input. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out1' and 'out2' are the same as in the list 2) Check that the loaded objects are the same as the saved objects. 3) The outputs 'out1' and 'out2' must be the same except.	pass
05	Tests that the save method works with a mix of different shaped TIMESPAN objects as input.	Test that the save method works with an input of matrices and vectors and single TIMESPAN objects. Test both formats 'xml' and 'mat'.	pass
		1) Check that the number of el- ements in 'out' is the same as in input. 2) Check that each output TIMESPAN object contains the correct data.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



università degli studi di trento

timespan/save			
06	Tests that the save method	Test that the result of apply-	pass
00	properly applies history.	ing the save method can be pro-	
		cessed back to an m-file. Do this	
		for both extensions 'mat' and	
		'xml'	
		1) Check that the history applies	pass
		to the output object. Check that	
		save doesn't add a history step to	
		the input object. 2) Check that	
		the read object doesn't contain	
		the save + load history steps.	
		3) Check that the method re-	
		build produces the same object	
		as 'out'.	
07	Tests that the save method	Use the save method with the	pass
07	works with the modify	modifier command.	
	command.	1) Check that the save method	pass
		doesn't apply the history. 2)	
		Check the output against the in-	
		put. 3) Check the history of the	
		output against the input.	
08	Control the method with a plict	Test that the save method uses	pass
00	Control the method with a plist.	the filename which is stored in a	
		plist.	
		1) Check the output	pass

Table 373: Unit tests for timespan/save.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)





Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

timespan/setEnd7			
07	Tests that the setEndT method can modify the input	Test that the setEndT method can modify the input TIMES-	pass
	TIMESPAN object.	PAN object by calling with no	
		output.	
		1) Check that 'ts5' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct name field	
08	Tests that the setEndT method	Test that the setEndT method	pass
	can set the property with a	can modify the property 'end'I'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect name field 2) Check that	
		the method rebuild produces the	
		same object as 'out'.	
09	Check that the setEnd1 method	Call the method with a list of	pass
	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
10	Check that the setEndT method	Test that the setEndT method	pass
	accept different inputs.	accept input of double, char and	
		time-objects.	
		1) Check the output 2) Check	pass
		that the 'rebuild' method pro-	
		duces the same object as 'out'.	

Table 374: Unit tests for timespan/setEndT.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



$  timespan/setN_{i}$	ame		
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
	Tests that the setName method	Test that the setName method	pass
02	works with a vector of	works for a vector of TIMESPAN	
	TIMESPAN objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	P
		'tsy' 2) Check that each output	
		contains the correct data	
	Tests that the setName method	Test that the setName method	nass
03	works with a matrix of	works for a matrix of TIMES-	pass
	TIMESPAN objects as input	PAN objects as input	
	TIMESTAR Objects as input.	1) Chock that the number of el	nage
		ements in 'out' is the same as in	pass
		'tam' 2) Chock that each output	
		contains the correct data	
	Tests that the setNeme method	Test that the astName method	<b>N</b> 2 2 2
04	ments with a list of TIMESDAN	rest that the setName method	pass
	works with a list of Thviespan	works for a list of TIMESPAN	
	objects as input.	1) Charle that the manufacture of all	
		1) Check that the number of el-	pass
		ements in out is the same as in	
		input. 2) Check that each output	
		contains the correct data.	
05	Tests that the setName method	Iest that the setName method	pass
	works with a mix of different	works with an input of matrices	
	shaped TIMESPAN objects as	and vectors and single TIMES-	
	input.	PAN objects.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		contains the correct data.	
06	Tests that the setName method	Test that the result of applying	pass
	properly applies history and	the setName method can be pro-	
	that the option 'internal'	cessed back to an m-file.	
	suppresses the history.	1) Check that the last entry in	pass
		the history of 'out1' corresponds	
		to 'setName'. 2) Check that the	
		last entry in the history of 'out2'	
		NOT corresponds to 'setName'.	
		3) Check that the method re-	
		build produces the same object	
		as 'out'.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

timespan/setName	e		
07	Tests that the setName method can modify the input	Test that the setName method can modify the input TIMES-	pass
	TIMESPAN object.	PAN object by calling with no	
		output.	
		1) Check that 'ts5' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct name field	
08	Tests that the setName method	Test that the setName method	pass
00	can set the property with a	can modify the property 'name'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect name field 2) Check that	
		the method rebuild produces the	
		same object as 'out'.	
09	Check that the setName method	Call the method with a list of	pass
05	pass back the output objects to	output variables and with a sin-	
	a list of output variables or to a	gle output variable. Additionaly	
	single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	

Table 375: Unit tests for timespan/setName.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



timespan/setStar	$\mathbf{t}$		
01	Tests that the getInfo call works	Test that the getInfo call works	pass
01	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
0.2	Tests that the setStartT method	Test that the setStartT method	pass
02	works with a vector of	works for a vector of TIMESPAN	
	TIMESPAN objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'tsv' 2) Check that each output	
		contains the correct data.	
0.2	Tests that the setStartT method	Test that the setStartT method	pass
03	works with a matrix of	works for a matrix of TIMES-	
	TIMESPAN objects as input.	PAN objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		'tsm' 2) Check that each output	
		contains the correct data.	
0.1	Tests that the setStartT method	Test that the setStartT method	pass
04	works with a list of TIMESPAN	works for a list of TIMESPAN	
	objects as input.	objects as input.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		contains the correct data.	
05	Tests that the setStartT method	Test that the setStartT method	pass
05	works with a mix of different	works with an input of matrices	
	shaped TIMESPAN objects as	and vectors and single TIMES-	
	input.	PAN objects.	
		1) Check that the number of el-	pass
		ements in 'out' is the same as in	
		input. 2) Check that each output	
		contains the correct data.	
0.0	Tests that the setStartT method	Test that the result of apply-	pass
06	properly applies history and	ing the setStartT method can be	
	that the option 'internal'	processed back to an m-file.	
	suppresses the history.	1) Check that the last entry in	pass
		the history of 'out1' corresponds	
		to 'setStartT'. 2) Check that the	
		last entry in the history of 'out2'	
		NOT corresponds to 'setStartT'.	
		3) Check that the method re-	
		build produces the same object	
		as 'out'.	



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

timespan/setStart	Т		
07	Tests that the setStartT method	Test that the setStartT method	pass
01	can modify the input	can modify the input TIMES-	
	TIMESPAN object.	PAN object by calling with no	
		output.	
		1) Check that 'ts5' and 'ain' are	pass
		now different. 2) Check that 'ain'	
		has the correct name field	
08	Tests that the setStartT method	Test that the setStartT method	pass
00	can set the property with a	can modify the property 'startT'	
	plist.	with a value in a plist.	
		1) Check that 'ain' has the cor-	pass
		rect name field 2) Check that	
		the method rebuild produces the	
		same object as 'out'.	
00	Check that the setStartT	Call the method with a list of	pass
09	method pass back the output	output variables and with a sin-	
	objects to a list of output	gle output variable. Additionaly	
	variables or to a single variable.	check that the rebuild method	
		works on the output.	
		1) Check that the output con-	pass
		tains the right number of ob-	
		jects 2) Check that the 'rebuild'	
		method produces the same ob-	
		ject as 'out'.	
10	Check that the setStartT	Test that the setStartT method	pass
10	method accept different inputs.	accept input of double, char and	
		time-objects.	
		1) Check the output 2) Check	pass
		that the 'rebuild' method pro-	
		duces the same object as 'out'.	

Table 376: Unit tests for timespan/setStartT.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

01Tests that the getInfo call works for this method.Test that the getInfo call works for no sets, all sets, and each set individually.011) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.02Tests that the string method works with a vector of TIMESPAN objects as input.Test that the string method works for a vector of TIMESPAN objects as input.011) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the	pass pass pass pass pass
02       Tests that the string method works with a vector of TIMESPAN objects as input.       Test that the string method works for a vector of TIMESPAN objects as input.         1)       Check that getInfo call returned an minfo object in all cases. 2)         02       Tests that the string method works with a vector of TIMESPAN objects as input.         1)       Check that all plists have the correct parameters.         10       Test that the string method works for a vector of TIMESPAN objects as input.         1)       Check that the output is a executable string. 2)         1)       Check the correct parameters.	pass pass pass pass
02       Tests that the string method works with a vector of TIMESPAN objects as input.       Test that the string method works for a vector of TIMESPAN objects as input.         1) Check that the output is a executable string. 2) Check the correct number of rout 3) Check the	pass pass pass
1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the	pass
rebuild objects.	pass
03Tests that the string method works with a matrix of TIMESPAN objects as input.Test that the string method works for a matrix of TIMES- PAN objects as input.	
1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
04 Tests that the string method works with a list of TIMESPAN objects as input. Test that the string method works for a list of TIMESPAN objects as input.	pass
1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
05 Tests that the string method works with a mix of different shaped TIMESPAN objects as input. Test that the string method works with an input of matrices and vectors and single TIMES- PAN objects.	pass
1) Check that the output is a ex- ecutable string. 2) Check the cor- rect number of rout 3) Check the rebuild objects.	pass
06 Tests that the string method properly applies history. The method string doesn't change the data, thus it is not possible to check the history. Nothing to do.	pass
07     Tests that the string method doesn't work if the TIMESPAN object have more than one history step.     The method string throws an er- ror because the input object have more than one history step.	pass pass

Table 377: Unit tests for timespan/string.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



## università degli studi di trento

timespan/times	pan		
00	Tests that the timespan constructor does what is supposed to do.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
01	Tests that the getInfo call works for this method.	Test that the getInfo call works for no sets, all sets, and each set individually.	pass
		1) Check that getInfo call re- turned an minfo object in all cases. 2) Check that all plists have the correct parameters.	pass
02	Tests that the timespan method works with a vector of TIMESPAN objects as input.	Test that the timespan method works with a vector of TIMES- PAN objects as input.	pass
		<ol> <li>Check that the shape of the output TIMESPANs is the same as the input shape.</li> <li>Check that each output TIMESPAN is a copy of the input TIMESPAN.</li> <li>Check that the copy have an additional history step.</li> </ol>	pass
03	Tests that the timespan method works with a matrix of TIMESPAN objects as input.	Test that the timespan method works with a matrix of TIMES- PAN objects as input.	pass
		<ol> <li>Check that the shape of the output TIMESPANs is the same as the input shape.</li> <li>Check that each output TIMESPAN is a copy of the input TIMESPAN.</li> <li>Check that the copy have an additional history step.</li> </ol>	pass
04	Tests that the timespan method works with a list of TIMESPAN objects as input.	Test that the timespan method works with a list of TIMESPAN objects as input.	pass
		<ol> <li>Check that the number of elements in 'out' is the same of the number in the input. 2) Check that each output TIMESPAN is a copy of the input TIMESPAN.</li> <li>Check that the copy have an additional history step.</li> </ol>	pass
05	Tests that the timespan method works with a mix of different shaped TIMESPANs as input.	Test that the timespan method works with a mix of different shaped TIMESPANs as input.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



timespan/times	pan		
		<ol> <li>Check that the number of elements in 'out' is the same of the number in the input. 2) Check that each output TIMESPAN is a copy of the input TIMESPAN.</li> <li>Check that the copy have an additional history step.</li> </ol>	pass
06	Tests that the timespan method properly applies history.	Test that the result of applying the timespan method can be pro- cessed back.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'timespan'. 2) Check that the method rebuild produces the same object as 'out'.	pass
07	Tests that the timespan method properly applies history to the copy constructor.	Test that the output can be pro- cessed back with the 'rebuild' method. Test the constructor with a different number of in- puts.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'timespan'. 2) Check that the original objects are not changed by the setter function 3) Check that the method rebuild pro- duces the same object as 'out'.	pass
08	Tests that the timespan method properly applies history to the read MAT-file constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		method produces the same object as 'out'.	I
09	Tests that the timespan method properly applies history to the read XML-file constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
10	Tests that the timespan method properly applies history to the struct constructor.	Test that the output can be pro- cessed back with the 'rebuild' method.	pass
		1) Check that the last entry in the history of 'out' corresponds to 'timespan'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
11	Tests that the timespan method properly applies history to the plist(filename) constructor.	Test that the output can be processed back to an m-file.	pass



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



## UNIVERSITÀ DEGLI STUDI DI TRENTO

timespan/timespan			
		1) Check that the save method doesn't change the input ob- ject 2) Check that the last two entries in the history of 'out' corresponds to 'timespan' and 'save' 3) Check that the 'rebuild' method produces the same ob- ject as 'out'.	pass
12	Tests that the TIMESPAN method properly applies history to the plist(conn) constructor.	<ul><li>1est that the output can be processed back with the rebuild method.</li><li>1) Check that the last entry in the history of 'out' corresponds to 'timespan'. 2) Check that the 'rebuild' method produces the same object as 'out'.</li></ul>	pass
13	Tests that the TIMESPAN method properly applies history to the plist(type) constructor.	Test that the output can be processed back with the rebuild method. 1) Check that the last entry in the history of 'out' corresponds to 'timespan'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
15	Tests that the TIMESPAN method properly applies history to the start + end time constructor.	Test that the output can be processed back with the rebuild method. 1) Check that the last entry in the history of 'out' corresponds to 'timespan'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
16	Tests that the TIMESPAN method properly applies history to the conn+Id constructor.	Test that the output can be processed back with the rebuild method. 1) Check that the last entry in the history of 'out' corresponds to 'timespan'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass
17	Tests that the TIMESPAN method properly applies history to the start + end time + format constructor.	Test that the output can be processed back with the rebuild method. 1) Check that the last entry in the history of 'out' corresponds to 'timespan'. 2) Check that the 'rebuild' method produces the same object as 'out'.	pass

Table 378: Unit tests for timespan/timespan.



Max Planck Institute for Gravitational Physics (Albert-Einstein-Institute)



UNIVERSITÀ DEGLI STUDI DI TRENTO

timespan/type			
01	Tests that the getInfo call works	Test that the getInfo call works	pass
	for this method.	for no sets, all sets, and each set	
		individually.	
		1) Check that getInfo call re-	pass
		turned an minfo object in all	
		cases. 2) Check that all plists	
		have the correct parameters.	
02	Tests that the type method	Test that the type method works	pass
	works with a vector of	for a vector of TIMESPAN ob-	
	TIMESPAN objects as input.	jects as input.	
		1) Check the rebuilt output.	pass
03	Tests that the type method	Test that the type method works	pass
	works with a matrix of	for a matrix of TIMESPAN ob-	
	TIMESPAN objects as input.	jects as input.	
		1) Check the rebuilt output.	pass
04	Tests that the type method	Test that the type method works	pass
	works with a list of TIMESPAN	for a list of TIMESPAN objects	
	objects as input.	as input.	
		1) Check the rebuilt output.	pass
05	Tests that the type method	Test that the type method works	pass
05	works with a mix of different	with an input of matrices and	
	shaped TIMESPAN objects as	vectors and single TIMESPAN	
	input.	objects.	
		1) Check the rebuilt output.	pass
06	Tests that the type method	The method type doesn't change	pass
	properly applies history.	the data, thus it is not possible	
		to check the history. Nothing to	
		do.	
			pass

Table 379: Unit tests for timespan/type.