Testing GlassBR

August 18, 2017

Table 1: testCalculations

Ref	Test Name	Old fileName.py	Test Purpose	Traceability	Input File	Significant Input	Expected Output	Notes
1	Tst_Pb_DefaultValues	testCalculations	to make sure expected pb values is returned	uses equations from DD1's B and IM1's Pb	defaultInput.txt	see Input File	'For the given input parameters, the glass is considered safe'	Improve: instead of equality of floats (assertEqual), should use some epsilon error
2	Tst_Pb_SmallDimensionValues	testCalculations2	"	"	testInput1.txt	"	"	"
3	Tst_Pb_LargeDimensionValues	testCalculations3	"	"	testInput2.txt	"	"	"
4	Tst_Pb_LowPbTol	testCalculations4	"	"	testInput3.txt	"	"	"
5	Tst_Pb_DiffSDValues	testCalculations5	"	"	testInput4.txt	"	"	"
6	Tst_Pb_HighChgWght	testCalculations6	"	"	testInput5.txt	"	"	"
7	Tst_Pb_LowThickness	testCalculations7	"	"	testInput6.txt	"	"	"

Table 2: testCheckConstraints

Ref	Test Name	Old fileName.py	Test Purpose	Traceability	Input File	Significant Input	Expected Output	Notes
8	checkAPositiveTest	testCheckConstraints	to ensure a (i.e. length) >0	Following A1 (glass must be of rectangular shape); following physical constraint from Table 2 where a >0 and software constraint from Table 2 where a =>d_{min}	testInvalidInput1.txt	a = -1600	InputError: a and b must be greater than 0	
9	checkBPositivwTest	testCheckConstraints2	to ensure b (i.e. breadth) >0	Following physical constraint from Table 2 where b >0 and software constraint from Table 2 where b =>d_{min}	testInvalidInput2.txt	b = -1500	InputError: a and b must be greater than 0	
10	checkSmallAspectRTest	testCheckConstraints3	to ensure 1 <a/b <5	length should pertain to the longer side, following physical constraint from Table 2 where b <a	testInvalidInput3.txt	b = 2000	(a/b=0.8<1); InputError: a/b must be between 1 and 5	
11	checkLargeAspectRTest	testCheckConstraints4	to ensure a/b (i.e. aspect ratio) <5	following software constraint from Table 2 where a/b <AR_{max}	testInvalidInput4.txt	b = 200	(a/b=8>5); InputError: a/b must be between 1 and 5	
12	checkValidThicknessTest	testCheckConstraints5	to ensure input t value (i.e. nominal thickness) is one of the industrial standard thicknesses	following R1 (t description)	testInvalidInput5.txt	t = 7	InputError: t must be in [2.5,2.7,3.0,4.0,,5.0,6.0,8.0, 10.0,12.0,16.0,,19.0,22.0]	
13	checkLowerConstrOnWTest	testCheckConstraints6	to ensure input w value (i.e. weight of charge) is >minimum permissible input charge weight	following value of w_{min} (4.5 kg) from Table 3	testInvalidInput6.txt	w = 3	InputError: wtnt must be between 4.5 and 910	
14	checkUpperConstrOnWTest	testCheckConstraints7	to ensure input w value (i.e. weight of charge) is <maximum permissible input charge weight	following value of w_{max} (910 kg) from Table 3	testInvalidInput7.txt	w = 1000	InputError: wtnt must be between 4.5 and 910	
15	checkTNTPositiveTest	testCheckConstraints8	to ensure input tnt value (i.e. TNT equivalent factor) >0	following physical constraint from Table 2 where TNT >0	testInvalidInput8.txt	tnt = -2	InputError: TNT must be greater than 0	
16	checkLowerConstrOnSDTest	testCheckConstraints9	to see if input SD (i.e. Stand off Distance) is >minimum stand off distance permissible for input	following value of SD_{min} (6 m) from Table 3	testInvalidInput9.txt	sdx = 0; sdy = 1.0; sdz = 2.0	InputError: SD must be between 6 and 130	
17	checkUpperConstrOnSDTest	testCheckConstraints10	to see if input SD (i.e. Stand off Distance) is <maximum stand off distance permissible for input	following value of SD_{max} (130 m) from Table 3	testInvalidInput10.txt	sdx = 0; sdy = 200; sdz = 100	InputError: SD must be between 6 and 130	
18	incorrectA0Test	testCheckConstraints11	see 8	see 8	testInvalidInput11.txt	a = 0	InputError: a and b must be greater than 0	
19	incorrectB0Test	testCheckConstraints12	see 9	see 9	testInvalidInput12.txt	b = 0	InputError: a and b must be greater than 0	RuntimeWarning: divide by zero encountered in double_scalars params.asprat = params.a /params.b
20	incorrectTNT0Test	testCheckConstraints13	see 15	see 15	testInvalidInput13.txt	tnt = 0	InputError: TNT must be greater than 0	
21	incorrectAspectREqLwrBndTest	testCheckConstraints14	see 10	see 10	testInput7.txt	a = 1500; b = 1500	(a/b = 1); "Encountered an unexpected exception"	why not the same error as 10?
22	incorrectAspectREqUpprBndTest	testCheckConstraints15	see 11	see 11	testInput8.txt	a = 7500; b = 1500	(a/b = 5); "Encountered an unexpected exception"	
23	incorrectWEqLwrBndTest	testCheckConstraints16	see 13	see 13	testInput9.txt	w = 4.5	"Encountered an unexpected exception"	
24	incorrectWEqUpprBndTest	testCheckConstraints17	see 14	see 14	testInput10.txt	w = 910	"Encountered an unexpected exception"	
25	incorrectSDEqLwrBndTest	testCheckConstraints18	see 16	see 16	testInput11.txt	sdx = 0; sdy = 6; sdz = 0	"Encountered an unexpected exception"	
26	incorrectWEqUpprBndTest	testCheckConstraints19	see 17	see 17	testInput12.txt	sdx = 130; sdy = 0; sdz = 0	"Encountered an unexpected exception"	

Table 3: testDerivedValues

Ref	Test Name	Old fileName.py	Test Purpose	Traceability	Input File	Significant Input	Expected Output	Notes
27	TstDrvdVals_HSGlTy	testDerivedValues	to ensure initial inputs have been correctly converted into derived quantities	following term definitions and equations from Data Definitions	defaultInput.txt	see Input File	-	
28	TstDrvdVals_ANGlTy	testDerivedValues2	"	"	testInput1.txt	"	-	
29	TstDrvdVals_FTGlTy	testDerivedValues3	"	"	testInput2.txt	"	-	

Table 4: testInputFormat

Ref	Test Name	Old fileName.py	Test Purpose	Traceability	Input File	Significant Input	Expected Output	Notes
30	TstInFmt_1	testInputFormat	to ensure data is being read in from the input file correctly	-	defaultInput.txt	see Input File	-	
31	TstInFmt_2	testInputFormat2	”	-	testInput1.txt	”	-	
32	TstInFmt_3	testInputFormat3	”	-	testInput2.txt	”	-	

Table 5: testInterp

Ref	Test Name	Old fileName.py	Test Purpose	Traceability	Input File	Significant Input	Expected Output	Notes
33	testInterp	testInterp	to ensure interpolated values are correctly calculated	-	-	x in [x1, x2]		
34	testInterp2	testInterp2	”	-	-	x in [x1, x2],where x is a Float		
35	testInterp3	testInterp3	”	-	-	x1 <x2 <x		
36	testInterp4	testInterp4	”	-	-	x <x2 <x1		
37	testInterp5	testInterp5	”	-	-	negative, x <x2 <x1		
38	testInterp6	testInterp6	”	-	-	negative, x in [x1, x2]		
39	testInterp7	testInterp7	”	-	-	negative, x1 <x2 <x		
40	testInterp8	testInterp8	”	-	-	value1 in data1 and jdx == 0?		
41	testInterp9	testInterp9	”	-	-	value1 in data1 and jdx == 0 and data2[jdx, idx] == data2[jdx+1, idx]		
42	testInterp10	testInterp10	”	-	-	value1 in data1 and data2[jdx, idx] == data2[jdx+1, idx] and value2 >data2[jdx, idx]		
43	testInterp11	testInterp11	”	-	-	value1 in data1 and data2[jdx, idx] == data2[jdx+1, idx] and value2 <data2[jdx, idx]		
44	testInterp12	testInterp12	”	-	-	value1 in data1 and data2[jdx, idx] == data2[jdx+1, idx] and jdx+1 == (data2[:, idx]).argmax and value2 >data2[jdx, idx]		
45	testInterp13	testInterp13	”	-	-	value1 not in data1 and value2 in data2[:, idx] and kdx == 0		
46	testInterp14	testInterp14	”	-	-	value1 not in data1 and value2 in data2[:, idx] and data2[:, idx+1] and data2[kdx, idx+1] = data2 [kdx+1, idx+1]		

Table 6: testMainFun

Ref	Test Name	Old fileName.py	Test Purpose	Traceability	Input File	Significant Input	Expected Output	Notes
47	TstMain_1	testMainFun	to determine if the main program produces the correct output	coordinates the running of the program	defaultInput.txt	see Input File	outputfile.txt = output.txt	
48	TstMain_2	testMainFun2	”	”	testInput1.txt	”	outputfile.txt = output1.txt	
49	TstMain_3	testMainFun3	”	”	testInput2.txt	”	outputfile.txt = output2.txt	
50	TstMain_4	testMainFun4	”	”	testInput3.txt	”	outputfile.txt = output3.txt	
51	TstMain_5	testMainFun5	”	”	testInput4.txt	”	outputfile.txt = output4.txt	
52	TstMain_6	testMainFun6	”	”	testInput5.txt	”	outputfile.txt = output5.txt	
53	TstMain_7	testMainFun7	”	”	testInput6.txt	”	outputfile.txt = output6.txt	

Table 7: testOutputFormat

Ref	Test Name	Old fileName.py	Test Purpose	Traceability	Input File	Significant Input	Expected Output	Notes
54	TstOutFmt_1	testOutputFormat	to ensure structure of output data matches desired output formatting	-	defaultInput.txt	see Input File	testoutput.txt = output.txt	
55	TstOutFmt_2	testOutputFormat2	”	-	testInput3.txt	”	testoutput.txt = output3.txt	
56	TstOutFmt_3	testOutputFormat3	”	-	testInput4.txt	”	testoutput.txt = output4.txt	

Table 8: testReadTable

Ref	Test Name	Old fileName.py	Test Purpose	Traceability	Input File	Significant Input	Expected Output	Notes
58	testReadTable	testReadTable	to make sure correct data is read from file (necessary for different interpolations)	-	testTable1.txt	see Input File		
59	testReadTable2	testReadTable2	”	-	testTable2.txt	”		