#	Equation	Alpha Strategy	Independent Variable Range	Correct Symmetries Found? *(considering all data from the first round of symmetry searching)* - Green = Perfect *(all possible symmetries found, if there are any, and no false symmetries found)* - Yellow = Partial *(some true symmetries found, but some true symmetries missed or false symmetries found)* - Red = Incorrect *(all symmetries missed, any symmetries found are false)*	# True Symmetries Found	# False Symmetries Found
1	(x0 - x1)**2 + (x2 - x3)**2 + (x4 - x5)**2 TRUE SYMMETRIES	Constant Alphas: - a_alpha: 5 - m_alpha: 2	-100 to 100	PARTIAL CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 1.604969493118613e-05) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0001024515872040110 4) [X] Symmetry Found: ((x4)-(x5)) -> (err = -4.177007355510298e-05) FALSE: [X] Symmetry Found: (((x0)-(x1))-((x2)-(x3))) -> (err = 0.005648359597501429)	3	1
			-50 to 50	PARTIAL CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = -0.00011760322009890523) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.00016337057956672396) [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.00012133814253989517) FALSE: [X] Symmetry Found: (((x0)-(x1))-((x2)-(x3))) -> (err = 0.025361276543598987)	3	1
			-10 to 10	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.002146723012388163) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0018536703322933334) [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.0012460824499216372)	3	0
			-1 to 1	INCORRECT MISSING: [X] Symmetry Found: ((x0)-(x1)) [X] Symmetry Found: ((x2)-(x3)) [X] Symmetry Found: ((x4)-(x5))	0	0
			-0.5 to 0.5	INCORRECT MISSING: [X] Symmetry Found: ((x0)-(x1)) [X] Symmetry Found: ((x2)-(x3)) [X] Symmetry Found: ((x4)-(x5))	0	0
		Adaptive by column	-100 to 100	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.004392841902108202) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.007890664437739225) [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.011844617630997756)	3	0
			-50 to 50	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.0075572559022184516) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.007241545252578474) [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.009534308358297272)	3	0
			-10 to 10	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.002907693840964898) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0018162788025187782) [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.0019246542047685056)	3	0
			-1 to 1	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.0007760008409021246) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0006221392552427929) [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.001039582584170251)	3	0

			-0.5 to 0.5	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.0011179597373045524) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0019527579670814976) [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.002199448127446546)	3	0
2	x0 * (x1 - x2)**2 TRUE SYMMETRIES	Constant Alphas: - a_alpha: 5 - m_alpha: 2	-100 to 100	INCORRECT FALSE: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.009283861868994103) MISSING: [X] Symmetry Found: ((x1)-(x2))	0	1
			-50 to 50	PERFECT CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.0008048932821270727)	1	0
			-10 to 10	INCORRECT MISSING: [X] Symmetry Found: ((x1)-(x2))	0	0
			-1 to 1	INCORRECT MISSING: [X] Symmetry Found: ((x1)-(x2))	0	0
			-0.5 to 0.5	INCORRECT MISSING: [X] Symmetry Found: ((x1)-(x2))	0	0
		Adaptive by column	-100 to 100	INCORRECT MISSING: [X] Symmetry Found: ((x1)-(x2))	0	0
			-50 to 50	INCORRECT MISSING: [X] Symmetry Found: ((x1)-(x2))	0	0
			-10 to 10	INCORRECT MISSING: [X] Symmetry Found: ((x1)-(x2))	0	0
			-1 to 1	PERFECT CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.00906434484512908)	1	0
			-0.5 to 0.5	PERFECT CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.005978021234848252)	1	0
	x0 * (x1 - x2)**2 + x3 * (x4 - x5)**2 TRUE SYMMETRIES	Constant Alphas: - a_alpha: 5 - m_alpha: 2	-100 to 100	PARTIAL CORRECT: [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.0006519901406225603) FALSE: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.005166658617538067) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.008009557812272527) MISSING: [X] Symmetry Found: ((x1)-(x2))	1	2

3			-50 to 50	PARTIAL CORRECT: [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.001112782423710712) FALSE: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.019109296377963103) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.021783375926681003) MISSING: [X] Symmetry Found: ((x1)-(x2))	1	2
			-10 to 10	PERFECT CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.006349973054349367) [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.004425531583548303)	2	0
			-1 to 1	INCORRECT MISSING: [X] Symmetry Found: ((x1)-(x2)) [X] Symmetry Found: ((x4)-(x5))	0	0
			-0.5 to 0.5	INCORRECT MISSING: [X] Symmetry Found: ((x1)-(x2)) [X] Symmetry Found: ((x4)-(x5))	0	0
		Adaptive by column	-100 to 100	PARTIAL CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.03148864975054999) FALSE: [X] Symmetry Found: ((x3)*(x4)) -> (err = 0.03539058568141695) MISSING: [X] Symmetry Found: ((x4)-(x5))	1	1
			-50 to 50	PERFECT CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.02767603183820122) [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.016587194684949957)	2	0
			-10 to 10	PERFECT CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.009906494649501663) [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.010439277712770645)	2	0
			-1 to 1	PERFECT CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.0007100187430252314) [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.001512830554261435)	2	0
			-0.5 to 0.5	PERFECT CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.0006893459229856402) [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.0006105249053344197)	2	0
	(x0-x1**2) * x2**3 FAKE SYMMETRIES	Constant Alphas: - a_alpha: 5 - m_alpha: 2	-100 to 100	INCORRECT FALSE: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.013701790709972683)	0	1
			-50 to 50	PERFECT No symmetries found	0	0
			-10 to 10	PERFECT No symmetries found	0	0

4			-1 to 1	PERFECT No symmetries found	0	0
			-0.5 to 0.5	PERFECT No symmetries found	0	0
		Adaptive by column	-100 to 100	PERFECT No symmetries found	0	0
			-50 to 50	PERFECT No symmetries found	0	0
			-10 to 10	PERFECT No symmetries found	0	0
			-1 to 1	PERFECT No symmetries found	0	0
			-0.5 to 0.5	PERFECT No symmetries found	0	0
5	(x0*x1) - (x2*x3) TRUE SYMMETRIES	Constant Alphas: - a_alpha: 5 - m_alpha: 2	-100 to 100	INCORRECT FALSE: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.004375229393929536) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.004186616634556883) MISSING: [X] Symmetry Found: ((x0)*(x1)) [X] Symmetry Found: ((x2)*(x3)) [X] Symmetry Found: (((x0)*(x1))-((x2)*(x3)))	0	2
			-50 to 50	INCORRECT FALSE: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.021685690066432062) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.024182865344863247) MISSING: [X] Symmetry Found: ((x0)*(x1)) [X] Symmetry Found: ((x2)*(x3)) [X] Symmetry Found: (((x0)*(x1))-((x2)*(x3)))	0	2
			-10 to 10	PERFECT CORRECT: [X] Symmetry Found: ((x0)*(x1)) -> (err = 0.014931601062250444) [X] Symmetry Found: ((x2)*(x3)) -> (err = 0.01726492787079048) [X] Symmetry Found: (((x0)*(x1))-((x2)*(x3))) -> (err = 2.1697846486201655e-07)	3	0
			-1 to 1	PARTIAL CORRECT: [X] Symmetry Found: ((x0)*(x1)) -> (err = 0.006866660200393393) [X] Symmetry Found: ((x2)*(x3)) -> (err = 0.007592186608872287) MISSING: [X] Symmetry Found: (((x0)*(x1))-((x2)*(x3)))	2	0
			-0.5 to 0.5	PARTIAL CORRECT: [X] Symmetry Found: ((x0)*(x1)) -> (err = 0.007159317122129916) [X] Symmetry Found: ((x2)*(x3)) -> (err = 0.005546864780809124) MISSING: [X] Symmetry Found: (((x0)*(x1))-((x2)*(x3)))	2	0

		Adaptive by column	-100 to 100	INCORRECT MISSING: [X] Symmetry Found: ((x0)*(x1)) [X] Symmetry Found: ((x2)*(x3)) [X] Symmetry Found: (((x0)*(x1))-((x2)*(x3)))	0	0
			-50 to 50	INCORRECT MISSING: [X] Symmetry Found: ((x0)*(x1)) [X] Symmetry Found: ((x2)*(x3)) [X] Symmetry Found: (((x0)*(x1))-((x2)*(x3)))	0	0
			-10 to 10	PERFECT CORRECT: [X] Symmetry Found: ((x0)*(x1)) -> (err = 0.002905150614727292) [X] Symmetry Found: ((x2)*(x3)) -> (err = 0.003681759438253218) [X] Symmetry Found: (((x0)*(x1))-((x2)*(x3))) -> (err = 1.2671354152615777e-06)	3	0
			-1 to 1	PERFECT CORRECT: [X] Symmetry Found: ((x0)*(x1)) -> (err = 0.002553059294648108) [X] Symmetry Found: ((x2)*(x3)) -> (err = 0.0032619437772480886) [X] Symmetry Found: (((x0)*(x1))-((x2)*(x3))) -> (err = 5.4755048070731505e-05)	3	0
			-0.5 to 0.5	PERFECT CORRECT: [X] Symmetry Found: ((x0)*(x1)) -> (err = 0.0030707200553867775) [X] Symmetry Found: ((x2)*(x3)) -> (err = 0.002394837147254081) [X] Symmetry Found: (((x0)*(x1))-((x2)*(x3))) -> (err = 0.00011401502055441437)	3	0
6	(x0 - x1)*(x2 - x3) TRUE SYMMETRIES	Constant Alphas: - a_alpha: 5 - m_alpha: 2	-100 to 100	PARTIAL CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.0008384231606212156) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0001159912550364739) MISSING: [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3)))	2	0
			-50 to 50	PARTIAL CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.0001720785415163517) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0001931121890870191 MISSING: [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3)))	2	0
			-10 to 10	PARTIAL CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.0017668918548879864) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0022730190103957826) MISSING: [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3)))	2	0
			-1 to 1	INCORRECT MISSING: [X] Symmetry Found: ((x0)-(x1)) [X] Symmetry Found: ((x2)-(x3)) [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3)))	0	0
			-0.5 to 0.5	INCORRECT MISSING: [X] Symmetry Found: ((x0)-(x1)) [X] Symmetry Found: ((x2)-(x3)) [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3)))	0	0

6		Adaptive by column	-100 to 100	INCORRECT MISSING: [X] Symmetry Found: ((x0)-(x1)) [X] Symmetry Found: ((x2)-(x3)) [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3)))	0	0
			-50 to 50	INCORRECT MISSING: [X] Symmetry Found: ((x0)-(x1)) [X] Symmetry Found: ((x2)-(x3)) [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3)))	0	0
			-10 to 10	PARTIAL CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.004588850761144125) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.002230025168201699) MISSING: [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3)))	2	0
			-1 to 1	PARTIAL CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.0011535527383385036) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.00066912520058382) MISSING: [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3)))	2	0
			-0.5 to 0.5	PARTIAL CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.0011637682304671548) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0008549106773731996) MISSING: [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3)))	2	0
	(x0 - x1)*(x2 - x3)*(x4 - x5) TRUE SYMMETRIES	Constant Alphas: - a_alpha: 5 - m_alpha: 2	-100 to 100	PARTIAL CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.00025180853258099933) [X] Symmetry Found: ((x2)-(x3)) -> (err = -0.0005667428069153768) [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.0017799100086284 27) MISSING: [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3))) [X] Symmetry Found: (((x4)-(x5))*(((x0)-(x1))*((x2)-(x3))))	3	0
			-50 to 50	PARTIAL CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.0029085635165959633) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0014450739957659886) [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.0019012314012633036) FALSE: [X] Symmetry Found: (((x0)-(x1))-((x2)-(x3))) -> (err = 0.025360712923110706) MISSING: [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3))) [X] Symmetry Found: (((x4)-(x5))*(((x0)-(x1))*((x2)-(x3))))	3	1
			-10 to 10	PARTIAL CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.007877404354170059) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.00809400575969721) [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.006447245356699383) MISSING: [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3))) [X] Symmetry Found: (((x4)-(x5))*(((x0)-(x1))*((x2)-(x3))))	3	0

7			-1 to 1	INCORRECT MISSING: [X] Symmetry Found: ((x0)-(x1)) [X] Symmetry Found: ((x2)-(x3)) [X] Symmetry Found: ((x4)-(x5)) [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3))) [X] Symmetry Found: (((x4)-(x5))*(((x0)-(x1))*((x2)-(x3))))	0	0
			-0.5 to 0.5	INCORRECT MISSING: [X] Symmetry Found: ((x0)-(x1)) [X] Symmetry Found: ((x2)-(x3)) [X] Symmetry Found: ((x4)-(x5)) [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3))) [X] Symmetry Found: (((x4)-(x5))*(((x0)-(x1))*((x2)-(x3))))	0	0
		Adaptive by column	-100 to 100	INCORRECT MISSING: [X] Symmetry Found: ((x0)-(x1)) [X] Symmetry Found: ((x2)-(x3)) [X] Symmetry Found: ((x4)-(x5)) [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3))) [X] Symmetry Found: (((x4)-(x5))*(((x0)-(x1))*((x2)-(x3))))	0	0
			-50 to 50	INCORRECT MISSING: [X] Symmetry Found: ((x0)-(x1)) [X] Symmetry Found: ((x2)-(x3)) [X] Symmetry Found: ((x4)-(x5)) [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3))) [X] Symmetry Found: (((x4)-(x5))*(((x0)-(x1))*((x2)-(x3))))	0	0
			-10 to 10	PARTIAL CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.01438692115723661) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.016030224194407028) [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.017656028486707553) MISSING: [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3))) [X] Symmetry Found: (((x4)-(x5))*(((x0)-(x1))*((x2)-(x3))))	3	0
			-1 to 1	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.0014643983547137918) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0023601046290858685) [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.0018474828137121646) [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3))) -> (err = 0.019925560254435037) [X] Symmetry Found: (((x4)-(x5))*(((x0)-(x1))*((x2)-(x3)))) -> (err = 0.01157072644144952)	5	0
			-0.5 to 0.5	PARTIAL CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.0028883669398279688) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.001216742325464315) [X] Symmetry Found: ((x4)-(x5)) -> (err = 0.002170690944837572) MISSING: [X] Symmetry Found: (((x0)-(x1))*((x2)-(x3))) [X] Symmetry Found: (((x4)-(x5))*(((x0)-(x1))*((x2)-(x3))))	3	0

8	x0*x1 - x2*(x3 + x4) TRUE SYMMETRIES	Constant Alphas: - a_alpha: 5 - m_alpha: 2	-100 to 100	INCORRECT FALSE: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.0033450973780898563) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.003066517217537701) MISSING: [X] Symmetry Found: ((x0)*(x1)) [X] Symmetry Found: ((x3)+(x4)) [X] Symmetry Found: ((x2)*((x3)+(x4))) [X] Symmetry Found: (((x0)*(x1))-((x2)*((x3)+(x4)))	0	2
			-50 to 50	INCORRECT FALSE: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.01906650338864968) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.020984919522028078) MISSING: [X] Symmetry Found: ((x0)*(x1)) [X] Symmetry Found: ((x3)+(x4)) [X] Symmetry Found: ((x2)*((x3)+(x4))) [X] Symmetry Found: (((x0)*(x1))-((x2)*((x3)+(x4)))	0	2
			-10 to 10	PERFECT CORRECT: [X] Symmetry Found: ((x0)*(x1)) -> (err = 0.01080160360004101) [X] Symmetry Found: ((x3)+(x4)) -> (err = 0.0015544455200185991) [X] Symmetry Found: ((x2)*((x3)+(x4))) -> (err = 0.012921399997343097) [X] Symmetry Found: (((x0)*(x1))-((x2)*((x3)+(x4)))) -> (err = 9.063938488651502e-08)	4	0
			-1 to 1	PARTIAL CORRECT: [X] Symmetry Found: ((x0)*(x1)) -> (err = 0.0024363907596607337) MISSING: [X] Symmetry Found: ((x3)+(x4)) [X] Symmetry Found: ((x2)*((x3)+(x4))) [X] Symmetry Found: (((x0)*(x1))-((x2)*((x3)+(x4)))	1	0
			-0.5 to 0.5	PARTIAL CORRECT: [X] Symmetry Found: ((x0)*(x1)) -> (err = 0.005048383978240256) MISSING: [X] Symmetry Found: ((x3)+(x4)) [X] Symmetry Found: ((x2)*((x3)+(x4))) [X] Symmetry Found: (((x0)*(x1))-((x2)*((x3)+(x4)))	1	0
		Adaptive by column	-100 to 100	PARTIAL CORRECT: [X] Symmetry Found: ((x0)*(x1)) -> (err = 0.035439850734505396) MISSING: [X] Symmetry Found: ((x3)+(x4)) [X] Symmetry Found: ((x2)*((x3)+(x4))) [X] Symmetry Found: (((x0)*(x1))-((x2)*((x3)+(x4)))	1	0
			-50 to 50	INCORRECT MISSING: [X] Symmetry Found: ((x0)*(x1)) [X] Symmetry Found: ((x3)+(x4)) [X] Symmetry Found: ((x2)*((x3)+(x4))) [X] Symmetry Found: (((x0)*(x1))-((x2)*((x3)+(x4)))	0	0

			-10 to 10	PERFECT CORRECT: [X] Symmetry Found: ((x0)*(x1)) -> (err = 0.0035324220568806464) [X] Symmetry Found: ((x3)+(x4)) -> (err = 0.0034536611403006567) [X] Symmetry Found: ((x2)*((x3)+(x4))) -> (err = 0.012921399997343097) [X] Symmetry Found: (((x0)*(x1))-((x2)*((x3)+(x4)))) -> (err = 9.063938488651502e-08)	4	0
			-1 to 1	PERFECT CORRECT: [X] Symmetry Found: ((x0)*(x1)) -> (err = 0.0007406767402375891) [X] Symmetry Found: ((x3)+(x4)) -> (err = 0.0005876952599708218) [X] Symmetry Found: ((x2)*((x3)+(x4))) -> (err = 0.012921399997343097) [X] Symmetry Found: (((x0)*(x1))-((x2)*((x3)+(x4)))) -> (err = 9.063938488651502e-08)	4	0
			-0.5 to 0.5	PERFECT CORRECT: [X] Symmetry Found: ((x0)*(x1)) -> (err = 0.0007075276856732815) [X] Symmetry Found: ((x3)+(x4)) -> (err = 0.0005190117547416717) [X] Symmetry Found: ((x2)*((x3)+(x4))) -> (err = 0.012921399997343097) [X] Symmetry Found: (((x0)*(x1))-((x2)*((x3)+(x4)))) -> (err = 9.063938488651502e-08)	4	0
9	(x0 - x1 + 5)**2 TRUE SYMMETRIES	Constant Alphas: - a_alpha: 5 - m_alpha: 2	-100 to 100	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 2.317398642148749e-05)	1	0
			-50 to 50	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.00035036930267373645)	1	0
			-10 to 10	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.005621590620392691)	1	0
			-1 to 1	INCORRECT MISSING: [X] Symmetry Found: ((x0)*(x1)) [X] Symmetry Found: ((x3)+(x4)) [X] Symmetry Found: ((x2)*((x3)+(x4))) [X] Symmetry Found: (((x0)*(x1))-((x2)*((x3)+(x4))))	0	0
			-0.5 to 0.5	INCORRECT MISSING: [X] Symmetry Found: ((x0)*(x1)) [X] Symmetry Found: ((x3)+(x4)) [X] Symmetry Found: ((x2)*((x3)+(x4))) [X] Symmetry Found: (((x0)*(x1))-((x2)*((x3)+(x4))))	0	0
		Adaptive by column	-100 to 100	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.023123166674517193)	1	0
			-50 to 50	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.024822882107659705)	1	0
			-10 to 10	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.006635673025090716)	1	0
			-1 to 1	INCORRECT MISSING: [X] Symmetry Found: ((x0)*(x1)) [X] Symmetry Found: ((x3)+(x4)) [X] Symmetry Found: ((x2)*((x3)+(x4))) [X] Symmetry Found: (((x0)*(x1))-((x2)*((x3)+(x4))))	0	0

			-0.5 to 0.5	INCORRECT MISSING: [X] Symmetry Found: ((x0)*(x1)) [X] Symmetry Found: ((x3)+(x4)) [X] Symmetry Found: ((x2)*((x3)+(x4))) [X] Symmetry Found: (((x0)*(x1))-((x2)*((x3)+(x4))))	0	0
10	(x0 - x1 + 5)**2 + (x2 - x3 - 10)**2 TRUE SYMMETRIES	Constant Alphas: - a_alpha: 5 - m_alpha: 2	-100 to 100	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 4.8346813595556704e-05) [X] Symmetry Found: ((x2)-(x3)) -> (err = 7.722039108415402e-05)	2	0
			-50 to 50	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.0002058504617064027 5) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0001142646838292460 4)	2	0
			-10 to 10	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.0009775320553108147) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0006728695200821644)	2	0
			-1 to 1	INCORRECT MISSING: [X] Symmetry Found: ((x0)-(x1)) [X] Symmetry Found: ((x2)-(x3))	0	0
			-0.5 to 0.5	INCORRECT MISSING: [X] Symmetry Found: ((x0)-(x1)) [X] Symmetry Found: ((x2)-(x3))	0	0
		Adaptive by column	-100 to 100	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.0142666600854010 06) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0124804642199545 23)	2	0
			-50 to 50	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.0132486520333054 97) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0089725871421521)	2	0
			-10 to 10	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.0026099946599499324) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0012635086616540336)	2	0
			-1 to 1	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.0026099946599499324) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0012635086616540336)	2	0
			-0.5 to 0.5	PERFECT CORRECT: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.0026099946599499324) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0012635086616540336)	2	0
	x0**2 + 2*x1*(x2-x3) PHYSICS - TRUE SYMMETRIES vf^2 = vi^2 + 2a(Δx)	Constant Alphas: - a_alpha: 5 - m_alpha: 2	-100 to 100	PARTIAL CORRECT: [X] Symmetry Found: ((x2)-(x3)) -> (err = 1.6762058005270752e-05) FALSE: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.00957180598463725) MISSING: [X] Symmetry Found: ((x1)*((x2)-(x3)))	1	1

11			-50 to 50	INCORRECT FALSE: [X] Symmetry Found: ((x0)-(x2)) -> (err = 0.02647892896970061) MISSING: [X] Symmetry Found: ((x2)-(x3)) [X] Symmetry Found: ((x1)*((x2)-(x3)))	0	1
			-10 to 10	PARTIAL CORRECT: [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0040010292636473155) MISSING: [X] Symmetry Found: ((x1)*((x2)-(x3)))	1	0
			-1 to 1	INCORRECT MISSING: [X] Symmetry Found: ((x2)-(x3)) [X] Symmetry Found: ((x1)*((x2)-(x3)))	0	0
			-0.5 to 0.5	INCORRECT MISSING: [X] Symmetry Found: ((x2)-(x3)) [X] Symmetry Found: ((x1)*((x2)-(x3)))	0	0
		Adaptive by column	-100 to 100	PARTIAL CORRECT: [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.007733589130265783) MISSING: [X] Symmetry Found: ((x1)*((x2)-(x3)))	1	0
			-50 to 50	PERFECT CORRECT: [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.006014395710193909) [X] Symmetry Found: ((x1)*((x2)-(x3))) -> (err = 0.01064572074584813)	2	0
			-10 to 10	PERFECT CORRECT: [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.00478044584879278) [X] Symmetry Found: ((x1)*((x2)-(x3))) -> (err = 0.009909772977650388)	2	0
			-1 to 1	PERFECT CORRECT: [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0016364381816914575) [X] Symmetry Found: ((x1)*((x2)-(x3))) -> (err = 0.004048413223665559)	2	0
			-0.5 to 0.5	PERFECT CORRECT: [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0014490571704159905) [X] Symmetry Found: ((x1)*((x2)-(x3))) -> (err = 0.004444821081199923)	2	0
	x0 * x1 * (x2 - x3) PHYSICS - TRUE SYMMETRIES ΔU = mgΔh	Constant Alphas: - a_alpha: 5 - m_alpha: 2	-100 to 100	PARTIAL CORRECT: [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.00025158978417061384) FALSE: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.016968247005879844) MISSING: [X] Symmetry Found: ((x0)*(x1)) [X] Symmetry Found: (((x0)*(x1))*((x2)-(x3)))	1	1

12			-50 to 50	PARTIAL CORRECT: [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0012907338248373534) FALSE: [X] Symmetry Found: ((x0)+((x2)-(x3))) -> (err = 0.030944588640936077) MISSING: [X] Symmetry Found: ((x0)*(x1)) [X] Symmetry Found: (((x0)*(x1))*((x2)-(x3)))	1	1
			-10 to 10	PARTIAL CORRECT: [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.0124049199990377) MISSING: [X] Symmetry Found: ((x0)*(x1)) [X] Symmetry Found: (((x0)*(x1))*((x2)-(x3)))	1	0
			-1 to 1	INCORRECT MISSING: [X] Symmetry Found: ((x0)*(x1)) [X] Symmetry Found: ((x2)-(x3)) [X] Symmetry Found: (((x0)*(x1))*((x2)-(x3)))	0	0
			-0.5 to 0.5	INCORRECT MISSING: [X] Symmetry Found: ((x0)*(x1)) [X] Symmetry Found: ((x2)-(x3)) [X] Symmetry Found: (((x0)*(x1))*((x2)-(x3)))	0	0
		Adaptive by column	-100 to 100	INCORRECT MISSING: [X] Symmetry Found: ((x0)*(x1)) [X] Symmetry Found: ((x2)-(x3)) [X] Symmetry Found: (((x0)*(x1))*((x2)-(x3)))	0	0
			-50 to 50	INCORRECT MISSING: [X] Symmetry Found: ((x0)*(x1)) [X] Symmetry Found: ((x2)-(x3)) [X] Symmetry Found: (((x0)*(x1))*((x2)-(x3)))	0	0
			-10 to 10	PARTIAL CORRECT: [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.01937529417824868) MISSING: [X] Symmetry Found: ((x0)*(x1)) [X] Symmetry Found: (((x0)*(x1))*((x2)-(x3)))	1	0
			-1 to 1	PARTIAL CORRECT: [X] Symmetry Found: ((x0)*(x1)) -> (err = 0.015121500212560646) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.008576784761932355) MISSING: [X] Symmetry Found: (((x0)*(x1))*((x2)-(x3)))	2	0
			-0.5 to 0.5	PARTIAL CORRECT: [X] Symmetry Found: ((x0)*(x1)) -> (err = 0.015121500212560646) [X] Symmetry Found: ((x2)-(x3)) -> (err = 0.008576784761932355) MISSING: [X] Symmetry Found: (((x0)*(x1))*((x2)-(x3)))	2	0

13	sin(x0*x1) PHYSICS - TRUE SYMMETRIES A = sin(ωt)	Constant Alphas: - a_alpha: 5 - m_alpha: 2	-100 to 100	INCORRECT MISSING: [X] Symmetry Found: ((x0)*(x1))	0	0
			-50 to 50	INCORRECT MISSING: [X] Symmetry Found: ((x0)*(x1))	0	0
			-10 to 10	INCORRECT MISSING: [X] Symmetry Found: ((x0)*(x1))	0	0
			-1 to 1	INCORRECT MISSING: [X] Symmetry Found: ((x0)*(x1))	0	0
			-0.5 to 0.5	INCORRECT MISSING: [X] Symmetry Found: ((x0)*(x1))	0	0
		Adaptive by column	-100 to 100	INCORRECT MISSING: [X] Symmetry Found: ((x0)*(x1))	0	0
			-50 to 50	INCORRECT MISSING: [X] Symmetry Found: ((x0)*(x1))	0	0
			-10 to 10	INCORRECT MISSING: [X] Symmetry Found: ((x0)*(x1))	0	0
			-1 to 1	PERFECT CORRECT: [X] Symmetry Found: ((x0)*(x1)) -> (err = 0.012697512874536021)	1	0
			-0.5 to 0.5	PERFECT CORRECT: [X] Symmetry Found: ((x0)*(x1)) -> (err = 0.017892991148388937)	1	0
14	0.5*x0*(x1**2 - x2**2) PHYSICS - FAKE SYMMETRIES ΔKE = 0.5m(vf^2 - vi^2)	Constant Alphas: - a_alpha: 5 - m_alpha: 2	-100 to 100	INCORRECT FALSE: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.018804216159643783)	0	1
			-50 to 50	PERFECT CORRECT: No symmetries found	0	0
			-10 to 10	PERFECT CORRECT: No symmetries found	0	0
			-1 to 1	PERFECT CORRECT: No symmetries found	0	0
			-0.5 to 0.5	PERFECT CORRECT: No symmetries found	0	0
		Adaptive by column	-100 to 100	PERFECT CORRECT: No symmetries found	0	0

			-50 to 50	PERFECT CORRECT: No symmetries found	0	0
			-10 to 10	PERFECT CORRECT: No symmetries found	0	0
			-1 to 1	PERFECT CORRECT: No symmetries found	0	0
			-0.5 to 0.5	PERFECT CORRECT: No symmetries found	0	0
15	(x0 - x1) + x1**2 + x2 TRUE SYMMETRIES	Constant Alphas: - a_alpha: 5 - m_alpha: 2	-100 to 100	INCORRECT FALSE: [X] Symmetry Found: ((x0)+(x1)) -> (err = 0.03583103652037396) MISSING: [X] Symmetry Found: ((x0)+(x2))	0	1
			-50 to 50	INCORRECT FALSE: [X] Symmetry Found: ((x0)-(x2)) -> (err = 0.0002290920445456867 7) MISSING: [X] Symmetry Found: ((x0)+(x2))	0	1
			-10 to 10	PERFECT CORRECT: [X] Symmetry Found: ((x0)+(x2)) -> (err = 0.0010364973374694664)	1	0
			-1 to 1	INCORRECT MISSING: [X] Symmetry Found: ((x0)+(x2))	0	0
			-0.5 to 0.5	INCORRECT MISSING: [X] Symmetry Found: ((x0)+(x2))	0	0
		Adaptive by column	-100 to 100	INCORRECT FALSE: [X] Symmetry Found: ((x0)-(x2)) -> (err = 0.006489872593555268) MISSING: [X] Symmetry Found: ((x0)+(x2))	0	1
			-50 to 50	INCORRECT FALSE: [X] Symmetry Found: ((x0)-(x2)) -> (err = 0.005905247104403877) MISSING: [X] Symmetry Found: ((x0)+(x2))	0	1
			-10 to 10	PERFECT CORRECT: [X] Symmetry Found: ((x0)+(x2)) -> (err = 0.0027675512742497688)	1	0
			-1 to 1	PERFECT CORRECT: [X] Symmetry Found: ((x0)+(x2)) -> (err = 0.0005931819231106417)	1	0

			-0.5 to 0.5	PERFECT CORRECT: [X] Symmetry Found: ((x0)+(x2)) -> (err = 0.0007387503821031371)	1	0
16	x0*(x1 - x2) TRUE SYMMETRIES	Constant Alphas: - a_alpha: 5 - m_alpha: 2	-100 to 100	INCORRECT FALSE: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.010911326733329396) MISSING: [X] Symmetry Found: ((x1)-(x2)) [X] Symmetry Found: ((x0)*((x1)-(x2)))	0	1
			-50 to 50	INCORRECT FALSE: [X] Symmetry Found: ((x0)+(x1)) -> (err = 0.031004224616896336) MISSING: [X] Symmetry Found: ((x1)-(x2)) [X] Symmetry Found: ((x0)*((x1)-(x2)))	0	1
			-10 to 10	PARTIAL CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.004419963418492379) MISSING: [X] Symmetry Found: ((x0)*((x1)-(x2)))	1	0
			-1 to 1	INCORRECT MISSING: [X] Symmetry Found: ((x1)-(x2)) [X] Symmetry Found: ((x0)*((x1)-(x2)))	0	0
			-0.5 to 0.5	INCORRECT MISSING: [X] Symmetry Found: ((x1)-(x2)) [X] Symmetry Found: ((x0)*((x1)-(x2)))	0	0
		Adaptive by column	-100 to 100	INCORRECT MISSING: [X] Symmetry Found: ((x1)-(x2)) [X] Symmetry Found: ((x0)*((x1)-(x2)))	0	0
			-50 to 50	PERFECT CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.0032596329637222254) [X] Symmetry Found: ((x0)*((x1)-(x2))) -> (err = 0.008352397311296111)	2	0
			-10 to 10	PERFECT CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.003148867085934093) [X] Symmetry Found: ((x0)*((x1)-(x2))) -> (err = 0.013221395383019585)	2	0
			-1 to 1	PERFECT CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.002179950029203459) [X] Symmetry Found: ((x0)*((x1)-(x2))) -> (err = 0.007861873547710951)	2	0
			-0.5 to 0.5	PERFECT CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.0016777605754975111) [X] Symmetry Found: ((x0)*((x1)-(x2))) -> (err = 0.009417925522741899)	2	0

17	x0 + x0**2 + x1 - x2 TRUE SYMMETRIES	Constant Alphas: - a_alpha: 5 - m_alpha: 2	-100 to 100	INCORRECT FALSE: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.03459270531961556) MISSING: [X] Symmetry Found: ((x1)-(x2))	0	1
			-50 to 50	PERFECT CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 3.688451414118887e-05)	1	0
			-10 to 10	PERFECT CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.0010199512749472506)	1	0
			-1 to 1	INCORRECT MISSING: [X] Symmetry Found: ((x1)-(x2))	0	0
			-0.5 to 0.5	INCORRECT MISSING: [X] Symmetry Found: ((x1)-(x2))	0	0
		Adaptive by column	-100 to 100	PERFECT CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.00822384941209009)	1	0
			-50 to 50	PERFECT CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.004659880057254373)	1	0
			-10 to 10	PERFECT CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.0020663673839489416)	1	0
			-1 to 1	PERFECT CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.0009186814425594658)	1	0
			-0.5 to 0.5	PERFECT CORRECT: [X] Symmetry Found: ((x1)-(x2)) -> (err = 0.00039208371403920417)	1	0
18	x0*(x0-x1-5) FAKE SYMMETRIES	Constant Alphas: - a_alpha: 5 - m_alpha: 2	-100 to 100	INCORRECT FALSE: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.008112192173037713)	0	1
			-50 to 50	INCORRECT FALSE: [X] Symmetry Found: ((x0)-(x1)) -> (err = 0.03170101761401023)	0	1
			-10 to 10	PERFECT CORRECT: No symmetries found	0	0
			-1 to 1	PERFECT CORRECT: No symmetries found	0	0
			-0.5 to 0.5	PERFECT CORRECT: No symmetries found	0	0

18		Adaptive by column	-100 to 100	PERFECT CORRECT: No symmetries found	0	0
			-50 to 50	PERFECT CORRECT: No symmetries found	0	0
			-10 to 10	PERFECT CORRECT: No symmetries found	0	0
			-1 to 1	PERFECT CORRECT: No symmetries found	0	0
			-0.5 to 0.5	PERFECT CORRECT: No symmetries found	0	0

Total	Constant	Adaptive
True Symmetries Found	60	110
False Symmetries Found	29	3

-100 to 100	Constant	Adaptive
True Symmetries Found	14	10
False Symmetries Found	16	2

-50 to 50	Constant	Adaptive
True Symmetries Found	15	13
False Symmetries Found	13	1

-10 to 10	Constant	Adaptive
True Symmetries Found	25	27
False Symmetries Found	0	0

-1 to 1	Constant	Adaptive
True Symmetries Found	3	31
False Symmetries Found	0	0

-0.5 to 0.5	Constant	Adaptive
True Symmetries Found	3	29
False Symmetries Found	0	0

Total	True Symmetries Found	False Symmetries Found
Constant	60	29
Adaptive	110	3

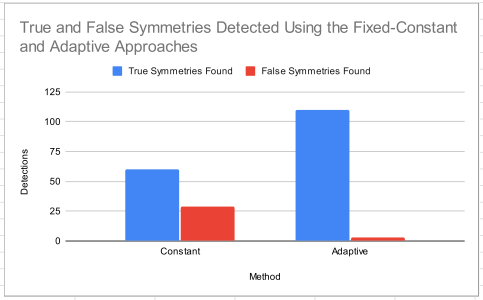
-100 to 100	True Symmetries Found	False Symmetries Found
Constant	14	16
Adaptive	10	2

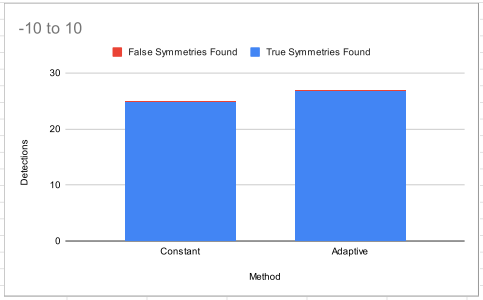
-50 to 50	True Symmetries Found	False Symmetries Found
Constant	15	13
Adaptive	13	1

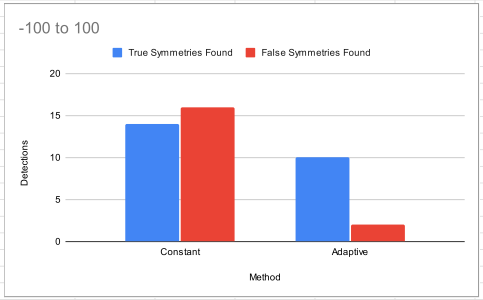
-10 to 10	True Symmetries Found	False Symmetries Found
Constant	25	0
Adaptive	27	0

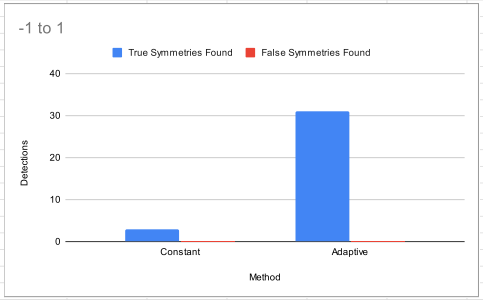
-1 to 1	True Symmetries Found	False Symmetries Found
Constant	3	0
Adaptive	31	0

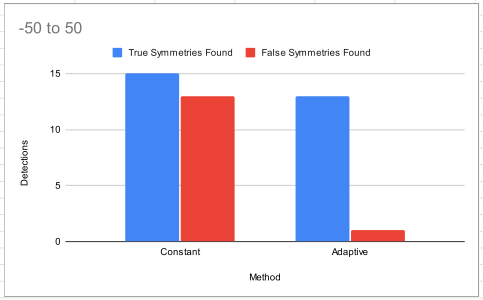
-0.5 to 0.5	True Symmetries Found	False Symmetries Found
Constant	3	0
Adaptive	29	0

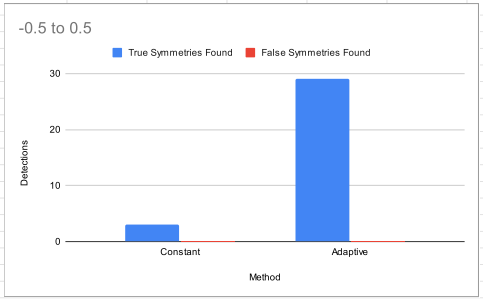












#	Equation Name	Equation	Noise Level (Standard Deviation of Normal Distribution)	Solution Found?	R^2
1	TME = Ug + Us + K	x0 + x1 + x2	0	1.0*(x0) + 1.0*(x1) + 1.0*(x2) + -0.0	1
			0.2	0.9994*(x0) + 0.9996*(x1) + 0.9991*(x2) + -0.0008	0.9984290554
			0.4	0.9951*(x0) + 0.9927*(x1) + 0.9938*(x2) + 0.0035	0.9934167759
			0.6	0.6082	0.6848276792
2	KE = 1/2 * m * v^2	0.5 * x0 * x1**2	0	0.5*((x0)*(x1)*(x1)) + -0.0	1
			0.2	0.4969*((x0)*(x1)*(x1)) + 0.3133	0.996113591
			0.4	17.0459	0.04548996413
			0.6	8.0422	0.4639633553
3	Ug = m*g*h	x0 * x1 * x2	0	1.0*((x0)*(x1)*(x2)) + -0.0	1
			0.2	0.9956*((x0)*(x1)*(x2)) + 0.2788	0.9963503747
			0.4	7.2795	0.004628772659
			0.6	1.4682*((((x0))-((x1)))-(((x2))-((x0)*(x0)))) + 3.6704*(sin(((x0)*(x0)*(x1)*(x1))-((x0)*(x0)*(x1)*(x2)))) + 0.8092*(sin(((x0)*(x0)*(x2)*(x2))-((x0)*(x1)*(x1)*(x1)))) + -2.3759*(sin(((x0)*(x1)*(x1)*(x2))+((x0)*(x1)*(x2)*(x2)))) + -1.1642*(sin(((x0)*(x2)*(x2)*(x2))-((x1)*(x1)*(x1)*(x1)))) + 2.2555*(sin(((x1)*(x1)*(x1)*(x2))-((x1)*(x1)*(x2)*(x2)))) + 3.0341*(sin(((x1)*(x2)*(x2)*(x2))-((x2)*(x2)*(x2)*(x2)))) + -2.1653*(sin((((x0))-((x1)))-(((x2))-((x0)*(x0))))) + 1.9655*(sin((((x0)*(x1))-((x0)*(x2)))-(((x1)*(x1))-((x1)*(x2))))) + 4.1892*(sin((((x2)*(x2))-((x0)*(x0)*(x0)))*(((x0)*(x2)*(x2))-((x1)*(x1)*(x1))))) + -5.1442*(sin((((x0)*(x0)*(x1))-((x0)*(x0)*(x2)))+(((x0)*(x1)*(x1))-((x0)*(x1)*(x2))))) + -6.6109*(sin((((x1)*(x1)*(x2))-((x1)*(x2)*(x2)))-(((x2)*(x2)*(x2))-((x0)*(x0)*(x0)*(x0))))) + -15.577	0.1584056318
4	xf = 1/2at2^2 + vit + xi	1/2 *x0*x1**2 + x2*x1 + x3	0	0.4963*((x0)*(x1)*(x1)) + 0.3203*(sin((x0)*(x0))) + 0.4024*(sin((x0)*((x2)*(x3)))) + -0.5576*(sin((x1)*(x1))) + -0.9697*(sin(((x2)*(x3))*((x2)*(x3)))) + 0.4194*(sin((x0)*(x0)*(x0))) + -0.3667*(sin((x0)*(x1)*(x1))) + -0.6957*(sin((x0)*((x2)*(x3))*((x2)*(x3)))) + -0.3911*(sin(((x2)*(x3))*((x2)*(x3))*((x2)*(x3)))) + 0.4913*(sin((x0)*(x0)*(x0)*(x0))) + -0.7309*(sin((x0)*(x0)*(x0)*((x2)*(x3)))) + 0.4402*(sin((x0)*(x1)*(x1)*(x1))) + 0.3592*(sin((x0)*(x1)*((x2)*(x3))*((x2)*(x3)))) + -0.323*(sin((x1)*(x1)*(x1)*(x1))) + 0.9879*(sin((x1)*(x1)*((x2)*(x3))*((x2)*(x3)))) + -1.8013	0.9363950561
			0.2	0.4954*((x0)*(x1)*(x1)) + 0.4235*(sin((x0))) + -1.1106*(sin((x1))) + -0.3196*(sin((x0)*(x0))) + 0.6809*(sin((x1)*(x1))) + 0.4118*(sin((x1)*((x2)*(x3)))) + -0.5603*(sin(((x2)*(x3))*((x2)*(x3)))) + -0.317*(sin((x0)*(x0)*(x0))) + -0.3412*(sin((x0)*(x0)*(x1))) + 0.3781*(sin((x0)*(x0)*((x2)*(x3)))) + 0.807*(sin((x0)*(x1)*((x2)*(x3)))) + 0.3385*(sin((x0)*((x2)*(x3))*((x2)*(x3)))) + -0.5835*(sin(((x2)*(x3))*((x2)*(x3))*((x2)*(x3)))) + 0.367*(sin((x0)*(x0)*(x0)*(x0))) + 0.7666*(sin((x0)*(x0)*(x1)*(x1))) + 0.5991*(sin((x0)*(x0)*((x2)*(x3))*((x2)*(x3)))) + -0.8119*(sin((x0)*(x1)*(x1)*(x1))) + 0.6482*(sin((x0)*(x1)*(x1)*((x2)*(x3)))) + 1.118*(sin((x0)*(x1)*((x2)*(x3))*((x2)*(x3)))) + 0.6654*(sin((x0)*((x2)*(x3))*((x2)*(x3))*((x2)*(x3)))) + 1.0525*(sin((x1)*(x1)*(x1)*(x1))) + -0.377*(sin((x1)*(x1)*((x2)*(x3))*((x2)*(x3)))) + -0.4188*(sin(((x2)*(x3))*((x2)*(x3))*((x2)*(x3))*((x2)*(x3)))) + -1.8475	0.9330469001

			0.4	0.4851*((x0)) + -0.3594*((x1)) + 0.4886*((x0)*(x1)*(x1)) + 0.5143*(sin((x0))) + -1.045*(sin((x1))) + -0.5186*(sin((x0)*((x2)*(x3)))) + 0.4439*(sin((x1)*((x2)*(x3)))) + -1.144*(sin(((x2)*(x3))*((x2)*(x3)))) + -0.7158*(sin((x0)*(x0)*(x0))) + -1.0791*(sin((x0)*(x0)*(x1))) + 0.4099*(sin((x0)*((x2)*(x3))*((x2)*(x3)))) + 0.774*(sin((x1)*(x1)*(x1))) + 1.236*(sin((x0)*(x0)*(x0)*(x1))) + 0.4514*(sin((x0)*(x0)*(x1)*(x1))) + 1.2106*(sin((x0)*(x0)*(x1)*((x2)*(x3)))) + 1.0769*(sin((x0)*(x1)*((x2)*(x3))*((x2)*(x3)))) + -0.912*(sin((x0)*((x2)*(x3))*((x2)*(x3))*((x2)*(x3)))) + 0.5442*(sin((x1)*(x1)*(x1)*((x2)*(x3)))) + 0.4738*(sin((x1)*((x2)*(x3))*((x2)*(x3))*((x2)*(x3)))) + 0.3525*(sin(((x2)*(x3))*((x2)*(x3))*((x2)*(x3))*((x2)*(x3)))) + 0.2541	0.9233915046
			0.6	1.4984*((x0)) + 0.4704*((x0)*(x1)*(x1)) + 1.0426*(sin((x0))) + -0.7658*(sin((x0)*(x1))) + 0.5164*(sin((x1)*(x1))) + 0.7919*(sin((x1)*((x2)*(x3)))) + -0.6322*(sin(((x2)*(x3))*((x2)*(x3)))) + 0.6028*(sin((x0)*(x0)*(x0))) + -0.633*(sin((x0)*(x0)*(x1))) + -0.9841*(sin((x0)*(x0)*((x2)*(x3)))) + 0.8905*(sin((x0)*(x1)*((x2)*(x3)))) + 0.3196*(sin((x0)*((x2)*(x3))*((x2)*(x3)))) + 0.5467*(sin((x1)*(x1)*(x1))) + -1.5416*(sin((x1)*((x2)*(x3))*((x2)*(x3)))) + 0.8659*(sin(((x2)*(x3))*((x2)*(x3))*((x2)*(x3)))) + 1.0733*(sin((x0)*(x0)*(x0)*(x0))) + 0.521*(sin((x0)*(x0)*(x0)*((x2)*(x3)))) + -0.471*(sin((x0)*(x0)*((x2)*(x3))*((x2)*(x3)))) + -0.575*(sin((x0)*(x1)*(x1)*((x2)*(x3)))) + 0.8705*(sin((x0)*(x1)*((x2)*(x3))*((x2)*(x3)))) + 1.6492*(sin((x0)*((x2)*(x3))*((x2)*(x3))*((x2)*(x3)))) + 0.513*(sin((x1)*(x1)*(x1)*(x1))) + -0.7672*(sin((x1)*((x2)*(x3))*((x2)*(x3))*((x2)*(x3)))) + -0.3207*(sin(((x2)*(x3))*((x2)*(x3))*((x2)*(x3))*((x2)*(x3)))) + -0.0864	0.9069318814
5	vf = at + vi	x0*x1 + x2	0	1.0*((x2)+((x0)*(x1))) + 0.0	1
			0.2	1.002*((x2)) + 1.0054*((x0)*(x1)) + 0.1207	0.9975548864
			0.4	0.9951*((x2)) + 1.0209*((x0)*(x1)) + 0.0897	0.9904110036
			0.6	-0.3302*(sin((((x0)*(x0)*(x1)*(x2))-((x0)*(x0)*(x2)*(x2)))-(((x1)*(x1)*(x1)*(x1))+((x1)*(x1)*(x1)*(x2))))) + -0.2951	0.8463173251
6	Power = F * d / t	x0 * (x1 - x2) / (x3 + 1e-5)	0	-2.1273*((x0)) + -4.7548*((x1)) + 2.1568*((x2)) + -0.513*((x3)) + -2.1585*((x0)*(x0)) + 2.0583*((x0)*(x1)) + 0.4582*((x0)*(x2)) + 0.8826*((x0)*(x3)) + -1.9982*((x1)*(x1)) + -0.5394*((x1)*(x2)) + -7.6949*((x2)*(x2)) + -0.4278*((x2)*(x3)) + -0.7595*((x3)*(x3)) + -3.2542*(1/((x0))) + -0.7802*(1/((x1))) + -5.8253*(1/((x2))) + 11.433*(1/((x3))) + 0.4802*(1/((x0)*(x1))) + 6.9372*(1/((x0)*(x2))) + 11.7501*(1/((x0)*(x3))) + 9.5657*(1/((x1)*(x2))) + 8.5358*(1/((x1)*(x3))) + 29.334*(1/((x2)*(x3))) + -7.1635*(1/((x0)*(x1)*(x2))) + 0.7229*(1/((x0)*(x1)*(x3))) + 1.4577*(1/((x0)*(x2)*(x3))) + -0.6082*(1/((x0)*(x3)*(x3))) + 2.0355*(1/((x1)*(x2)*(x3))) + 0.5881*(1/((x2)*(x3)*(x3))) + 159.8425	0.9945176346
			0.2	6.0659*((x0)) + -2.5416*((x3)) + -1.6272*((x0)*(x0)) + 0.8051*((x0)*(x2)) + -0.3494*((x0)*(x3)) + -1.9795*((x1)*(x1)) + 0.6371*((x1)*(x2)) + -1.536*((x2)*(x2)) + -1.2404*((x3)*(x3)) + -0.6161*((x0)*(x0)*(x0)) + 0.3658*((x2)*(x3)*(x3)) + 4.7608*(1/((x0))) + -4.311*(1/((x1))) + 3.5131*(1/((x2))) + 22.6095*(1/((x3))) + 4.7889*(1/((x0)*(x1))) + -3.8952*(1/((x0)*(x2))) + -6.3945*(1/((x0)*(x3))) + -1.4939*(1/((x1)*(x2))) + -13.857*(1/((x1)*(x3))) + 13.538*(1/((x2)*(x3))) + -0.4205*(1/((x0)*(x0)*(x1))) + -4.7254*(1/((x0)*(x1)*(x2))) + 21.5761*(1/((x0)*(x1)*(x3))) + -0.483*(1/((x0)*(x2)*(x2))) + -4.0101*(1/((x0)*(x2)*(x3))) + 0.5917*(1/((x1)*(x1)*(x2))) + -0.5424*(1/((x1)*(x1)*(x3))) + 11.7005*(1/((x1)*(x2)*(x3))) + -0.8666*(1/((x0)*(x0)*(x1)*(x2))) + 0.7791*(1/((x0)*(x0)*(x2)*(x3))) + 0.8322*(1/((x0)*(x1)*(x1)*(x2))) + -0.8101*(1/((x0)*(x1)*(x1)*(x3))) + -1.189*(1/((x0)*(x1)*(x2)*(x2))) + 2.4545*(1/((x0)*(x1)*(x2)*(x3))) + -0.6063*(1/((x0)*(x1)*(x3)*(x3))) + 0.4568*(1/((x0)*(x2)*(x2)*(x3))) + 0.4516*(1/((x0)*(x2)*(x3)*(x3))) + -0.518*(1/((x1)*(x2)*(x3)*(x3))) + 97.3554	-0.3846153994

			0.4	9.9189*((x0)) + 23.142*((x1)) + -23.4275*((x2)) + -19.2475*((x3)) + 2.1303*((x0)*(x1)) + -0.4856*((x0)*(x2)) + 0.4709*((x1)*(x1)) + -1.8858*((x1)*(x2)) + 0.7669*((x2)*(x2)) + -0.5185*((x2)*(x3)) + 4.3802*(sin((x0))) + -2.1952*(sin((x1))) + 1.2867*(sin((x2))) + -0.5917*(sin((x3))) + 2.5*(sin((x0)*(x0))) + -9.7028*(sin((x0)*(x1))) + 7.1229*(sin((x0)*(x2))) + 5.329*(sin((x0)*(x3))) + 13.8867*(sin((x1)*(x1))) + 9.419*(sin((x1)*(x2))) + 12.1334*(sin((x1)*(x3))) + -4.6611*(sin((x2)*(x2))) + -5.2754*(sin((x2)*(x3))) + 0.4864*(sin((x3)*(x3))) + -1.7827*(sin((x0)*(x0)*(x0))) + -6.9634*(sin((x0)*(x0)*(x1))) + 6.3694*(sin((x0)*(x0)*(x2))) + 5.7544*(sin((x0)*(x0)*(x3))) + 1.2539*(sin((x0)*(x1)*(x1))) + 0.8605*(sin((x0)*(x1)*(x2))) + 6.7646*(sin((x0)*(x1)*(x3))) + -3.0449*(sin((x0)*(x2)*(x2))) + -1.5134*(sin((x0)*(x2)*(x3))) + -1.0403*(sin((x0)*(x3)*(x3))) + 8.5088*(sin((x1)*(x1)*(x1))) + -4.6944*(sin((x1)*(x1)*(x2))) + 12.8097*(sin((x1)*(x1)*(x3))) + 4.9809*(sin((x1)*(x2)*(x2))) + -9.1956*(sin((x1)*(x2)*(x3))) + 1.2471*(sin((x1)*(x3)*(x3))) + -8.2118*(sin((x2)*(x2)*(x2))) + 11.0873*(sin((x2)*(x2)*(x3))) + -3.4913*(sin((x2)*(x3)*(x3))) + -0.3274*(sin((x3)*(x3)*(x3))) + 9.9995*(sin((x0)*(x0)*(x0)*(x0))) + -8.4682*(sin((x0)*(x0)*(x0)*(x1))) + 4.0763*(sin((x0)*(x0)*(x0)*(x2))) + -8.6002*(sin((x0)*(x0)*(x0)*(x3))) + -9.8431*(sin((x0)*(x0)*(x1)*(x1))) + -9.2166*(sin((x0)*(x0)*(x1)*(x2))) + -4.0735*(sin((x0)*(x0)*(x1)*(x3))) + 3.6924*(sin((x0)*(x0)*(x2)*(x2))) + 7.4423*(sin((x0)*(x0)*(x2)*(x3))) + 7.0604*(sin((x0)*(x1)*(x1)*(x1))) + -2.6943*(sin((x0)*(x1)*(x1)*(x2))) + 12.6404*(sin((x0)*(x1)*(x1)*(x3))) + -3.694*(sin((x0)*(x1)*(x2)*(x2))) + -1.6134*(sin((x0)*(x1)*(x2)*(x3))) + -6.9594*(sin((x0)*(x1)*(x3)*(x3))) + 5.9055*(sin((x0)*(x2)*(x2)*(x2))) + -4.7899*(sin((x0)*(x2)*(x2)*(x3))) + 5.616*(sin((x0)*(x2)*(x3)*(x3))) + 0.5975*(sin((x0)*(x3)*(x3)*(x3))) + 4.1797*(sin((x1)*(x1)*(x1)*(x1))) + -3.2141*(sin((x1)*(x1)*(x1)*(x2))) + 8.5617*(sin((x1)*(x1)*(x2)*(x2))) + -4.9292*(sin((x1)*(x1)*(x2)*(x3))) + -1.4964*(sin((x1)*(x2)*(x2)*(x2))) + -1.5109*(sin((x1)*(x2)*(x2)*(x3))) + 14.8307*(sin((x1)*(x2)*(x3)*(x3))) + 4.7722*(sin((x1)*(x3)*(x3)*(x3))) + -2.6351*(sin((x2)*(x2)*(x2)*(x2))) + 4.7455*(sin((x2)*(x2)*(x2)*(x3))) + -3.7712*(sin((x2)*(x2)*(x3)*(x3))) + -5.6059*(sin((x2)*(x3)*(x3)*(x3))) + -2.1424*(1/((x1))) + 2.7964*(1/((x2))) + 1.0224*(1/((x3))) + 0.4084*(1/((x0)*(x2))) + -0.5402*(1/((x1)*(x3))) + -0.868*(1/((x2)*(x3))) + -0.3681*(1/((x0)*(x1)*(x1))) + 1.0249*(1/((x0)*(x1)*(x3))) + -0.8375*(1/((x0)*(x2)*(x3))) + -1.7068*(1/((x1)*(x2)*(x3))) + -0.3703*(1/((x0)*(x0)*(x1)*(x2))) + 0.3956*(1/((x0)*(x1)*(x2)*(x3))) + 84.2704	-0.4054740671

			0.6	7.363*((x0)) + -20.6811*((x1)) + 50.7791*((x2)) + -20.8925*((x3)) + 3.3029*((x0)*(x0)) + -1.8154*((x0)*(x1)) + 0.9121*((x0)*(x2)) + -0.4573*((x1)*(x1)) + 1.1012*((x2)*(x2)) + 0.6167*((x2)*(x3)) + -0.5306*((x0)*(x1)*(x2)) + 0.3501*((x1)*(x1)*(x3)) + -0.4195*((x2)*(x2)*(x2)) + 0.6092*((x3)*(x3)*(x3)) + -20.83*(sin((x0))) + -13.4543*(sin((x1))) + -1.099*(sin((x2))) + -2.4927*(sin((x3))) + -20.1602*(sin((x0)*(x0))) + 13.9198*(sin((x0)*(x1))) + 14.8797*(sin((x0)*(x2))) + -13.8264*(sin((x0)*(x3))) + 20.6273*(sin((x1)*(x1))) + 14.2287*(sin((x1)*(x2))) + 5.8743*(sin((x1)*(x3))) + 14.1995*(sin((x2)*(x2))) + -18.1014*(sin((x2)*(x3))) + 4.884*(sin((x3)*(x3))) + 2.0394*(sin((x0)*(x0)*(x0))) + -2.0716*(sin((x0)*(x0)*(x1))) + -2.4135*(sin((x0)*(x0)*(x2))) + -18.6073*(sin((x0)*(x0)*(x3))) + -26.2491*(sin((x0)*(x1)*(x1))) + -9.5296*(sin((x0)*(x1)*(x2))) + 23.8878*(sin((x0)*(x1)*(x3))) + 9.945*(sin((x0)*(x2)*(x2))) + 6.4179*(sin((x0)*(x2)*(x3))) + 20.5142*(sin((x0)*(x3)*(x3))) + 12.8262*(sin((x1)*(x1)*(x1))) + -5.7411*(sin((x1)*(x1)*(x2))) + 4.4133*(sin((x1)*(x1)*(x3))) + 5.0319*(sin((x1)*(x2)*(x2))) + 7.9788*(sin((x1)*(x2)*(x3))) + -11.5103*(sin((x1)*(x3)*(x3))) + -10.1833*(sin((x2)*(x2)*(x2))) + 0.859*(sin((x2)*(x2)*(x3))) + 12.1133*(sin((x2)*(x3)*(x3))) + 11.799*(sin((x3)*(x3)*(x3))) + -22.0062*(sin((x0)*(x0)*(x0)*(x0))) + 24.7653*(sin((x0)*(x0)*(x0)*(x1))) + 12.0444*(sin((x0)*(x0)*(x0)*(x2))) + 2.4245*(sin((x0)*(x0)*(x0)*(x3))) + -17.4381*(sin((x0)*(x0)*(x1)*(x1))) + -21.0332*(sin((x0)*(x0)*(x1)*(x2))) + 20.9598*(sin((x0)*(x0)*(x1)*(x3))) + 27.4833*(sin((x0)*(x0)*(x2)*(x2))) + 30.2988*(sin((x0)*(x0)*(x2)*(x3))) + 30.6692*(sin((x0)*(x0)*(x3)*(x3))) + 4.4835*(sin((x0)*(x1)*(x1)*(x1))) + 19.3862*(sin((x0)*(x1)*(x1)*(x2))) + -14.4641*(sin((x0)*(x1)*(x1)*(x3))) + 5.2009*(sin((x0)*(x1)*(x2)*(x2))) + 17.023*(sin((x0)*(x1)*(x2)*(x3))) + -12.974*(sin((x0)*(x1)*(x3)*(x3))) + -24.1604*(sin((x0)*(x2)*(x2)*(x2))) + -9.1248*(sin((x0)*(x2)*(x2)*(x3))) + -22.9049*(sin((x0)*(x2)*(x3)*(x3))) + -5.2325*(sin((x0)*(x3)*(x3)*(x3))) + 8.6662*(sin((x1)*(x1)*(x1)*(x1))) + -12.3201*(sin((x1)*(x1)*(x1)*(x2))) + -15.7505*(sin((x1)*(x1)*(x1)*(x3))) + 20.7156*(sin((x1)*(x1)*(x2)*(x2))) + 22.1429*(sin((x1)*(x1)*(x2)*(x3))) + -23.7098*(sin((x1)*(x1)*(x3)*(x3))) + 0.449*(sin((x1)*(x2)*(x2)*(x2))) + -21.3426*(sin((x1)*(x2)*(x2)*(x3))) + 7.1011*(sin((x1)*(x2)*(x3)*(x3))) + 7.849*(sin((x1)*(x3)*(x3)*(x3))) + 23.762*(sin((x2)*(x2)*(x2)*(x2))) + -15.6511*(sin((x2)*(x2)*(x2)*(x3))) + -7.2132*(sin((x2)*(x2)*(x3)*(x3))) + 16.6945*(sin((x2)*(x3)*(x3)*(x3))) + -11.5767*(sin((x3)*(x3)*(x3)*(x3))) + -6.6805*(1/((x0))) + 4.1887*(1/((x1))) + -1.2681*(1/((x2))) + 16.4716*(1/((x3))) + -0.3422*(1/((x0)*(x0))) + -6.2091*(1/((x0)*(x1))) + -5.1443*(1/((x0)*(x2))) + 8.0071*(1/((x0)*(x3))) + -2.269*(1/((x1)*(x1))) + -3.3403*(1/((x1)*(x3))) + -1.0814*(1/((x2)*(x3))) + 0.5546*(1/((x0)*(x0)*(x1))) + -0.3563*(1/((x0)*(x0)*(x3))) + 1.3712*(1/((x0)*(x1)*(x1))) + 2.1168*(1/((x0)*(x1)*(x2))) + 2.502*(1/((x0)*(x1)*(x3))) + 3.8108*(1/((x0)*(x2)*(x3))) + 0.8343*(1/((x1)*(x1)*(x2))) + 0.9048*(1/((x1)*(x1)*(x3))) + -1.2904*(1/((x1)*(x2)*(x2))) + 2.9247*(1/((x1)*(x2)*(x3))) + -1.5422*(1/((x2)*(x3)*(x3))) + -1.9701*(1/((x0)*(x0)*(x1)*(x2))) + 1.8552*(1/((x0)*(x0)*(x1)*(x3))) + 0.5266*(1/((x0)*(x0)*(x2)*(x3))) + -0.6009*(1/((x0)*(x0)*(x3)*(x3))) + 1.4726*(1/((x0)*(x1)*(x1)*(x2))) + -1.9892*(1/((x0)*(x1)*(x1)*(x3))) + -2.5395*(1/((x0)*(x1)*(x2)*(x2))) + 7.2174*(1/((x0)*(x1)*(x2)*(x3))) + 1.6089*(1/((x0)*(x1)*(x3)*(x3))) + -0.7651*(1/((x0)*(x2)*(x2)*(x3))) + 0.6274*(1/((x0)*(x2)*(x3)*(x3))) + 0.3344*(1/((x1)*(x1)*(x2)*(x2))) + 0.8971*(1/((x1)*(x1)*(x2)*(x3))) + -0.3409*(1/((x1)*(x2)*(x3)*(x3))) + -37.5726	-0.6745180795
7	Ue = 1/2 k (xf - xi)^2	1/2 * (x0) * (x1 - x2)**2	0	0.5*((x0)*(x1)*(x1)) + -1.0*((x0)*(x1)*(x2)) + 0.5*((x0)*(x2)*(x2)) + -0.0	1
			0.2	0.5098*((x0)) + 0.4937*((x0)*(x1)*(x1)) + -0.9953*((x0)*(x1)*(x2)) + 0.4959*((x0)*(x2)*(x2)) + -0.1606	0.9968081682
			0.4	0.347*(((((x0)*(x1)*(x1))-((x0)*(x1)*(x2)))+(((x0)*(x2)*(x2))-((x1)*(x1)*(x1))))-(sin((((x0)*(x1)*(x1))-((x0)*(x1)*(x2)))+(((x0)*(x2)*(x2))-((x1)*(x1)*(x1))))))) + 0.5444*((sin((((x1)*(x2)*(x2)*(x2))-((x2)*(x2)*(x2)*(x2)))-((((x0))-((x1)))-(((x2))-((x0)*(x0)))))-(sin(((((x0)*(x1))-((x0)*(x2)))-(((x1)*(x1))-((x1)*(x2))))-((((x2)*(x2))-((x0)*(x0)*(x0)))-(((x0)*(x0)*(x1))-((x0)*(x0)*(x2))))))) + 14.523	0.6442870639
			0.6	-0.3231*((1/((((((x0)*(x1))-((x0)*(x2)))-(((x1)*(x1))-((x1)*(x2))))-((((x2)*(x2))-((x0)*(x0)*(x0)))-(((x0)*(x0)*(x1))-((x0)*(x0)*(x2))))))-(((((x0)*(x1)*(x1))-((x0)*(x1)*(x2)))+(((x0)*(x2)*(x2))-((x1)*(x1)*(x1))))-(sin((((x0)*(x1)*(x1))-((x0)*(x1)*(x2)))+(((x0)*(x2)*(x2))-((x1)*(x1)*(x1))))))) + 4.444	0.6331920199

8	I = mr^2	x0 * x1**2	0	1.0*((x0)*(x1)*(x1)) + 0.0	1
			0.2	0.9952*((x0)*(x1)*(x1)) + -0.3633	0.996363959
			0.4	-0.3194*((((x1)*(x1))-((x0)*(x0)*(x0)))+(((x0)*(x0)*(x1))-((x0)*(x1)*(x1)))) + 3.9843*(sin(((x1)*(x1)*(x1))-((x0)*(x0)*(x0)*(x0)))) + -1.1692*(sin(((x0)*(x0)*(x0)*(x1))-((x0)*(x0)*(x1)*(x1)))) + 0.8009*(sin(((x0)*(x1)*(x1)*(x1))-(((x0))-((x1))))) + -1.7294*(sin(((x1)*(x1)*(x1)*(x1))-(((x0)*(x0))-((x0)*(x1))))) + -0.5483*(sin((((x1)*(x1))-((x0)*(x0)*(x0)))+(((x0)*(x0)*(x1))-((x0)*(x1)*(x1))))) + 9.3114	0.6278569897
			0.6	2.2024*(sin(((x1)*(x1)*(x1))-(((x0))-((x1))))) + 2.9843*(sin((((x0)*(x0)*(x1)*(x1))+((x1)*(x1)*(x1)*(x1)))-(((x0)*(x1)*(x1)*(x1))-(((x0)*(x0))-((x0)*(x1)))))) + 7.8152*(sin((((x1)*(x1))-((x0)*(x0)*(x0)))*((((x0)*(x0)*(x1))*((x0)*(x1)*(x1)))-(((x0)*(x0)*(x0)*(x0))+((x0)*(x0)*(x0)*(x1)))))) + -2.6825	0.001832536875
9	Fdrag = 1/2 * C * p * A * v^2	0.5 * x0 * x1**2 * x2 * x3	0	20.9677*(sin(((x3)*(x3)*(x3)*(x3))-(((x0))-((x1))))) + 25.0838*(sin((((x2)*(x2)*(x2))-((x2)*(x2)*(x3)))+(((x2)*(x3)*(x3))-((x3)*(x3)*(x3))))) + 65.082*(sin((((x0)*(x0)*(x0)*(x0))-((x0)*(x0)*(x0)*(x1)))-(((x0)*(x1)*(x1)*(x1))-((x0)*(x1)*(x1)*(x2))))) + -40.6419*(sin((((x0)*(x0)*(x0)*(x2))-((x0)*(x0)*(x0)*(x3)))+(((x0)*(x0)*(x1)*(x1))-((x0)*(x0)*(x1)*(x2))))) + 43.4149*(sin((((x0)*(x1)*(x2)*(x3))-((x0)*(x1)*(x3)*(x3)))+(((x0)*(x2)*(x2)*(x2))-((x0)*(x2)*(x2)*(x3))))) + -47.3223*(sin((((x0)*(x2)*(x3)*(x3))-((x0)*(x3)*(x3)*(x3)))-(((x1)*(x1)*(x1)*(x1))-((x1)*(x1)*(x1)*(x2))))) + -11.9693*(sin((((x1)*(x2)*(x2)*(x2))-((x1)*(x2)*(x2)*(x3)))+(((x2)*(x2)*(x2)*(x2))-((x2)*(x2)*(x2)*(x3))))) + 28.8718*(sin((((x0)*(x0)*(x1)*(x3))-((x0)*(x0)*(x2)*(x2)))+(((x1)*(x1)*(x1)*(x3))-((x1)*(x1)*(x2)*(x2))))) + 53.5379*(sin((((x0)*(x0)*(x2)*(x3))-((x0)*(x0)*(x3)*(x3)))-((((x2))-((x3)))-(((x0)*(x0))-((x0)*(x1)))))) + 34.5628*(sin((((x0)*(x1)*(x1)*(x3))-((x0)*(x1)*(x2)*(x2)))-((((x0)*(x2))-((x0)*(x3)))-(((x1)*(x1))-((x1)*(x2)))))) + -8.377*(sin((((x1)*(x1)*(x2)*(x3))-((x1)*(x1)*(x3)*(x3)))-((((x1)*(x3))-((x2)*(x2)))-(((x2)*(x3))-((x3)*(x3)))))) + -74.9607*(sin((((x1)*(x2)*(x3)*(x3))-((x1)*(x3)*(x3)*(x3)))-((((x0)*(x2)*(x3))-((x0)*(x3)*(x3)))-(((x1)*(x1)*(x1))-((x1)*(x1)*(x2)))))) + 64.7526*(sin((((x2)*(x2)*(x3)*(x3))-((x2)*(x3)*(x3)*(x3)))*((((x0)*(x1)*(x1))-((x0)*(x1)*(x2)))-(((x0)*(x1)*(x3))-((x0)*(x2)*(x2)))))) + 20.522*(sin(((((x0)*(x0)*(x0))-((x0)*(x0)*(x1)))+(((x0)*(x0)*(x2))-((x0)*(x0)*(x3))))*((((x1)*(x1)*(x3))-((x1)*(x2)*(x2)))-(((x1)*(x2)*(x3))-((x1)*(x3)*(x3)))))) + 0.8242*(1/(((x3)*(x3)*(x3)*(x3))-(((x0))-((x1))))) + 0.37*(1/((((x0)*(x1)*(x2)*(x3))-((x0)*(x1)*(x3)*(x3)))+(((x0)*(x2)*(x2)*(x2))-((x0)*(x2)*(x2)*(x3))))) + -0.9809*(1/((((x0)*(x2)*(x3)*(x3))-((x0)*(x3)*(x3)*(x3)))-(((x1)*(x1)*(x1)*(x1))-((x1)*(x1)*(x1)*(x2))))) + -3.9264*(1/((((x0)*(x0)*(x1)*(x3))-((x0)*(x0)*(x2)*(x2)))+(((x1)*(x1)*(x1)*(x3))-((x1)*(x1)*(x2)*(x2))))) + -5.5904*(1/((((x0)*(x1)*(x1)*(x3))-((x0)*(x1)*(x2)*(x2)))-((((x0)*(x2))-((x0)*(x3)))-(((x1)*(x1))-((x1)*(x2)))))) + -0.6425*(1/((((x1)*(x1)*(x2)*(x3))-((x1)*(x1)*(x3)*(x3)))-((((x1)*(x3))-((x2)*(x2)))-(((x2)*(x3))-((x3)*(x3)))))) + 6.0252*(1/((((x1)*(x2)*(x3)*(x3))-((x1)*(x3)*(x3)*(x3)))-((((x0)*(x2)*(x3))-((x0)*(x3)*(x3)))-(((x1)*(x1)*(x1))-((x1)*(x1)*(x2)))))) + -6.1261	0.3586249872

			0.2	33.104*(sin((((x0)*(x0)*(x0)*(x0))-((x0)*(x0)*(x0)*(x1)))-(((x0)*(x2)*(x3)*(x3))-((x0)*(x3)*(x3)*(x3))))) + 23.3634*(sin((((x0)*(x0)*(x1)*(x1))-((x0)*(x0)*(x1)*(x2)))+(((x1)*(x1)*(x1)*(x1))-((x1)*(x1)*(x1)*(x2))))) + -38.5579*(sin((((x1)*(x1)*(x1)*(x3))-((x1)*(x1)*(x2)*(x2)))-(((x1)*(x2)*(x2)*(x2))-((x1)*(x2)*(x2)*(x3))))) + -60.948*(sin((((x1)*(x1)*(x2)*(x3))-((x1)*(x1)*(x3)*(x3)))+(((x1)*(x2)*(x3)*(x3))-((x1)*(x3)*(x3)*(x3))))) + 27.155*(sin((((x0)*(x0)*(x0)*(x2))-((x0)*(x0)*(x0)*(x3)))*(((x0)*(x1)*(x2)*(x3))-((x0)*(x1)*(x3)*(x3))))) + -67.7707*(sin((((x0)*(x0)*(x1)*(x3))-((x0)*(x0)*(x2)*(x2)))-((((x2))-((x3)))-(((x0)*(x0))-((x0)*(x1)))))) + -35.5927*(sin((((x0)*(x0)*(x2)*(x3))-((x0)*(x0)*(x3)*(x3)))-((((x0)*(x2))-((x0)*(x3)))-(((x1)*(x1))-((x1)*(x2)))))) + 43.6946*(sin((((x0)*(x1)*(x1)*(x1))-((x0)*(x1)*(x1)*(x2)))+(((x2)*(x2)*(x2)*(x2))-((x2)*(x2)*(x2)*(x3))))) + -38.9457*(sin((((x0)*(x1)*(x1)*(x3))-((x0)*(x1)*(x2)*(x2)))-((((x1)*(x3))-((x2)*(x2)))-(((x2)*(x3))-((x3)*(x3)))))) + 154.3809*(sin((((x0)*(x2)*(x2)*(x2))-((x0)*(x2)*(x2)*(x3)))*((((x0)*(x1)*(x1))-((x0)*(x1)*(x2)))-(((x0)*(x1)*(x3))-((x0)*(x2)*(x2)))))) + 81.8403*(sin((((x2)*(x2)*(x3)*(x3))-((x2)*(x3)*(x3)*(x3)))-((((x0)*(x2)*(x3))-((x0)*(x3)*(x3)))-(((x1)*(x1)*(x1))-((x1)*(x1)*(x2)))))) + -8.4699*(sin((((x3)*(x3)*(x3)*(x3))-(((x0))-((x1))))-((((x2)*(x2)*(x2))-((x2)*(x2)*(x3)))-(((x2)*(x3)*(x3))-((x3)*(x3)*(x3)))))) + 39.3941*(sin(((((x0)*(x0)*(x0))-((x0)*(x0)*(x1)))-(((x0)*(x0)*(x2))-((x0)*(x0)*(x3))))*((((x1)*(x1)*(x3))-((x1)*(x2)*(x2)))-(((x1)*(x2)*(x3))-((x1)*(x3)*(x3)))))) + -0.7601*(1/((((x1)*(x1)*(x1)*(x3))-((x1)*(x1)*(x2)*(x2)))-(((x1)*(x2)*(x2)*(x2))-((x1)*(x2)*(x2)*(x3))))) + 2.0172*(1/((((x0)*(x0)*(x2)*(x3))-((x0)*(x0)*(x3)*(x3)))-((((x0)*(x2))-((x0)*(x3)))-(((x1)*(x1))-((x1)*(x2)))))) + 0.8519*(1/((((x0)*(x1)*(x1)*(x1))-((x0)*(x1)*(x1)*(x2)))+(((x2)*(x2)*(x2)*(x2))-((x2)*(x2)*(x2)*(x3))))) + -7.4776*(1/((((x0)*(x1)*(x1)*(x3))-((x0)*(x1)*(x2)*(x2)))-((((x1)*(x3))-((x2)*(x2)))-(((x2)*(x3))-((x3)*(x3)))))) + 4.2173*(1/((((x2)*(x2)*(x3)*(x3))-((x2)*(x3)*(x3)*(x3)))-((((x0)*(x2)*(x3))-((x0)*(x3)*(x3)))-(((x1)*(x1)*(x1))-((x1)*(x1)*(x2)))))) + -1.9141*(1/(((x3)*(x3)*(x3)*(x3))-(((x0))-((x1))))-((((x2)*(x2)*(x2))-((x2)*(x2)*(x3)))-(((x2)*(x3)*(x3))-((x3)*(x3)*(x3)))))) + -186.361	0.01052684635
			0.4	-20.6371*(sin((((x0)*(x0)*(x0)*(x0))-((x0)*(x0)*(x0)*(x1)))+(((x1)*(x2)*(x2)*(x2))-((x1)*(x2)*(x2)*(x3))))) + -116.6375*(sin((((x0)*(x0)*(x0)*(x2))-((x0)*(x0)*(x0)*(x3)))-(((x0)*(x0)*(x1)*(x1))-((x0)*(x0)*(x1)*(x2))))) + -37.4188*(sin((((x0)*(x0)*(x1)*(x3))-((x0)*(x0)*(x2)*(x2)))+(((x2)*(x2)*(x3)*(x3))-((x2)*(x3)*(x3)*(x3)))))) + 72.3185*(sin((((x0)*(x1)*(x1)*(x3))-((x0)*(x1)*(x2)*(x2)))-(((x0)*(x2)*(x2)*(x2))-((x0)*(x2)*(x2)*(x3))))) + -42.638*(sin((((x0)*(x2)*(x3)*(x3))-((x0)*(x3)*(x3)*(x3)))+(((x1)*(x1)*(x1)*(x1))-((x1)*(x1)*(x1)*(x2))))) + -20.2253*(sin((((x1)*(x2)*(x3)*(x3))-((x1)*(x3)*(x3)*(x3)))-(((x2)*(x2)*(x2)*(x2))-((x2)*(x2)*(x2)*(x3))))) + 118.6202*(sin((((x0)*(x0)*(x2)*(x3))-((x0)*(x0)*(x3)*(x3)))-((((x2))-((x3)))-(((x0)*(x0))-((x0)*(x1)))))) + -41.0768*(sin((((x0)*(x1)*(x1)*(x1))-((x0)*(x1)*(x1)*(x2)))-(((x0)*(x1)*(x2)*(x3))-((x0)*(x1)*(x3)*(x3))))) + -74.6683*(sin((((x1)*(x1)*(x1)*(x3))-((x1)*(x1)*(x2)*(x2)))-((((x0)*(x2))-((x0)*(x3)))-(((x1)*(x1))-((x1)*(x2)))))) + 18.0092*(sin((((x1)*(x1)*(x2)*(x3))-((x1)*(x1)*(x3)*(x3)))-((((x1)*(x3))-((x2)*(x2)))-(((x2)*(x3))-((x3)*(x3)))))) + -9.4476*(sin((((x3)*(x3)*(x3)*(x3))-(((x0))-((x1))))+((((x0)*(x1)*(x1))-((x0)*(x1)*(x2)))-(((x0)*(x1)*(x3))-((x0)*(x2)*(x2)))))) + -182.385*(sin(((((x0)*(x0)*(x0))-((x0)*(x0)*(x1)))+(((x0)*(x0)*(x2))-((x0)*(x0)*(x3))))*((((x0)*(x2)*(x3))-((x0)*(x3)*(x3)))-(((x1)*(x1)*(x1))-((x1)*(x1)*(x2)))))) + 9.2393*(sin(((((x1)*(x1)*(x3))-((x1)*(x2)*(x2)))-(((x1)*(x2)*(x3))-((x1)*(x3)*(x3))))-((((x2)*(x2)*(x2))-((x2)*(x2)*(x3)))+(((x2)*(x3)*(x3))-((x3)*(x3)*(x3)))))) + -3.8714*(1/((((x0)*(x0)*(x0)*(x0))-((x0)*(x0)*(x0)*(x1)))+(((x1)*(x2)*(x2)*(x2))-((x1)*(x2)*(x2)*(x3))))) + -0.8307*(1/((((x0)*(x0)*(x1)*(x3))-((x0)*(x0)*(x2)*(x2)))+(((x2)*(x2)*(x3)*(x3))-((x2)*(x3)*(x3)*(x3))))) + 1.4748*(1/((((x0)*(x1)*(x1)*(x3))-((x0)*(x1)*(x2)*(x2)))-(((x0)*(x2)*(x2)*(x2))-((x0)*(x2)*(x2)*(x3))))) + -3.185*(1/((((x0)*(x2)*(x3)*(x3))-((x0)*(x3)*(x3)*(x3)))+(((x1)*(x1)*(x1)*(x1))-((x1)*(x1)*(x1)*(x2))))) + -8.1494*(1/((((x0)*(x0)*(x2)*(x3))-((x0)*(x0)*(x3)*(x3)))-((((x2))-((x3)))-(((x0)*(x0))-((x0)*(x1)))))) + -0.3215*(1/((((x0)*(x1)*(x1)*(x1))-((x0)*(x1)*(x1)*(x2)))-(((x0)*(x1)*(x2)*(x3))-((x0)*(x1)*(x3)*(x3))))) + 0.6927*(1/((((x1)*(x1)*(x1)*(x3))-((x1)*(x1)*(x2)*(x2)))-((((x0)*(x2))-((x0)*(x3)))-(((x1)*(x1))-((x1)*(x2)))))) + 0.6345*(1/((((x1)*(x1)*(x2)*(x3))-((x1)*(x1)*(x3)*(x3)))-((((x1)*(x3))-((x2)*(x2)))-(((x2)*(x3))-((x3)*(x3)))))) + 59.9092*(1/((((x3)*(x3)*(x3)*(x3))-(((x0))-((x1))))+((((x0)*(x1)*(x1))-((x0)*(x1)*(x2)))-(((x0)*(x1)*(x3))-((x0)*(x2)*(x2)))))) + 0.9448*(1/(((((x1)*(x1)*(x3))-((x1)*(x2)*(x2)))-(((x1)*(x2)*(x3))-((x1)*(x3)*(x3))))-((((x2)*(x2)*(x2))-((x2)*(x2)*(x3)))+(((x2)*(x3)*(x3))-((x3)*(x3)*(x3)))))) + -75.6848	0.005232629372

			0.6	-91.7834*((sin(((x1)*(x1)*(x1)*(x3))-((x1)*(x1)*(x2)*(x2))))-(sin(((x1)*(x1)*(x2)*(x3))-((x1)*(x1)*(x3)*(x3))))) + 18.3757*((sin(((x1)*(x2)*(x3)*(x3))-((x1)*(x3)*(x3)*(x3))))-(sin(((x2)*(x2)*(x2)*(x2))-((x2)*(x2)*(x2)*(x3))))) + 49.5739*((sin(((x2)*(x2)*(x3)*(x3))-((x2)*(x3)*(x3)*(x3))))-(sin(((x3)*(x3)*(x3)*(x3))-(((x0))-((x1)))))) + -74.4496*((sin((((x2))-((x3)))-(((x0)*(x0))-((x0)*(x1)))))-(sin((((x0)*(x2))-((x0)*(x3)))-(((x1)*(x1))-((x1)*(x2)))))) + -257.3586*((sin((((x1)*(x3))-((x2)*(x2)))-(((x2)*(x3))-((x3)*(x3)))))-(sin((((x0)*(x0)*(x0))-((x0)*(x0)*(x1)))-(((x0)*(x0)*(x2))-((x0)*(x0)*(x3)))))) + -74.1089*((sin((((x0)*(x1)*(x1))-((x0)*(x1)*(x2)))+(((x0)*(x1)*(x3))-((x0)*(x2)*(x2)))))-(sin((((x0)*(x2)*(x3))-((x0)*(x3)*(x3)))+(((x1)*(x1)*(x1))-((x1)*(x1)*(x2)))))) + 40.2936*((sin((((x1)*(x1)*(x3))-((x1)*(x2)*(x2)))-(((x1)*(x2)*(x3))-((x1)*(x3)*(x3)))))-(sin((((x2)*(x2)*(x2))-((x2)*(x2)*(x3)))+(((x2)*(x3)*(x3))-((x3)*(x3)*(x3)))))) + -109.5667*((sin((((x0)*(x0)*(x0)*(x0))-((x0)*(x0)*(x0)*(x1)))-(((x0)*(x0)*(x1)*(x1))-((x0)*(x0)*(x1)*(x2)))))-(sin((((x0)*(x0)*(x0)*(x2))-((x0)*(x0)*(x0)*(x3)))-(((x0)*(x1)*(x1)*(x3))-((x0)*(x1)*(x2)*(x2)))))) + 62.8968*((sin((((x0)*(x2)*(x2)*(x2))-((x0)*(x2)*(x2)*(x3)))+(((x1)*(x1)*(x1)*(x1))-((x1)*(x1)*(x1)*(x2)))))-(sin((((x0)*(x2)*(x3)*(x3))-((x0)*(x3)*(x3)*(x3)))+(((x1)*(x2)*(x2)*(x2))-((x1)*(x2)*(x2)*(x3)))))) + -3.0811*((1/((((x2))-((x3)))-(((x0)*(x0))-((x0)*(x1)))))-(1/((((x0)*(x2))-((x0)*(x3)))-(((x1)*(x1))-((x1)*(x2)))))) + -14.1001*((1/((((x0)*(x1)*(x1))-((x0)*(x1)*(x2)))+(((x0)*(x1)*(x3))-((x0)*(x2)*(x2)))))-(1/((((x0)*(x2)*(x3))-((x0)*(x3)*(x3)))+(((x1)*(x1)*(x1))-((x1)*(x1)*(x2)))))) + 0.941*((1/((((x1)*(x1)*(x3))-((x1)*(x2)*(x2)))-(((x1)*(x2)*(x3))-((x1)*(x3)*(x3)))))-(1/((((x2)*(x2)*(x2))-((x2)*(x2)*(x3)))+(((x2)*(x3)*(x3))-((x3)*(x3)*(x3)))))) + 40.4724	0.006333489826
10	W = F * (x2 - x1)	x0*(x1-x2)	0	1.0*((x0)*((x1)-(x2))) + 0.0	1
			0.2	0.9969*((x0)*((x1)-(x2))) + -0.0282	0.9975357109
			0.4	1.0158*((x0)*((x1)-(x2))) + -0.0359	0.9904126723
			0.6	0.6378*(sin((x0)*(x0)*(x0)*((x1)-(x2)))) + 1.841*(sin((x0)*((x1)-(x2))*((x1)-(x2))*((x1)-(x2)))) + -0.9073*(sin((((x1)-(x2))*((x1)-(x2)))-((x0)*(x0)*(x0)))) + 1.2366*(sin(((x0)*(x0)*((x1)-(x2)))+(((x1)-(x2))*((x1)-(x2))*((x1)-(x2))))) + -1.092*(sin(((x0)*((x1)-(x2))*((x1)-(x2)))-(((x0))-(((x1)-(x2)))))) + 3.5132*(sin(((x0)*(x0)*(x0)*(x0))-(((x0)*(x0))-((x0)*((x1)-(x2)))))) + 0.5971*(1/((((x1)-(x2))*((x1)-(x2)))-((x0)*(x0)*(x0)))) + -25.5892	-0.03310934398
11	j = sigma * T^4	x0 * x1**4	0	-21.3723*(((((x1)*(x1))-((x0)*(x0)*(x0)))+(((x0)*(x0)*(x1))-((x0)*(x1)*(x1))))-((sin(((x0)*(x0)*(x1))-((x0)*(x1)*(x1))))-(sin(((x1)*(x1)*(x1))-((x0)*(x0)*(x0)*(x0)))))) + 172.7741	0.5031019116
			0.2	-21.1703*(((((x1)*(x1))-((x0)*(x0)*(x0)))+(((x0)*(x0)*(x1))-((x0)*(x1)*(x1))))-(sin((((x1)*(x1))-((x0)*(x0)*(x0)))+(((x0)*(x0)*(x1))-((x0)*(x1)*(x1)))))) + 621.287	0.5038398841
			0.4	-20.7081*(((((x1)*(x1))-((x0)*(x0)*(x0)))+(((x0)*(x0)*(x1))-((x0)*(x1)*(x1))))-(sin(((x0)*(x0)*(x1)*(x1))+((x1)*(x1)*(x1)*(x1))))) + 361.3643	0.493500965
			0.6	-291.7568	0.003569542779
12	P = n*R*T/V	x0 * x1 * x2 / x3	0	-1.3285*((x0)*(x0)) + 1.8549*((x1)) + 0.3453*((x2)) + -0.4867*((x2)*(x2)) + 0.7254*((x3)*(x3)) + -3.5847*(sin((x0))) + -5.5965*(sin((x0)*(x0))) + -6.3586*(sin((x0)*(x0)*(x0))) + -17.0717*(sin((x0)*(x0)*(x0)*(x0))) + -8.9074*(sin((x1))) + -5.6549*(sin((x1)*(x1))) + -21.8037*(sin((x1)*(x1)*(x1))) + -0.7102*(sin((x1)*(x1)*(x1)*(x1))) + -0.6321*(sin((x2))) + 9.935*(sin((x2)*(x2))) + -4.6699*(sin((x2)*(x2)*(x2))) + -11.2529*(sin((x2)*(x2)*(x2)*(x2))) + 0.5173*(sin((x3))) + 0.7303*(sin((x3)*(x3))) + -0.7584*(sin((x3)*(x3)*(x3))) + -1.4896*(sin((x3)*(x3)*(x3)*(x3))) + 1.8242	0.9973504534

			0.2	-34.4271*((x0)) + 56.6389*((x1)) + -45.3773*((x2)) + -17.5187*((x3)) + -71.3395*((x0)*(x0)) + 39.0564*((x0)*(x1)) + 11.5525*((x0)*(x2)) + -12.989*((x0)*(x3)) + -60.766*((x1)*(x1)) + 1.8274*((x1)*(x2)) + -19.2292*((x1)*(x3)) + 8.2483*((x2)*(x2)) + -42.4123*((x2)*(x3)) + -13.1229*((x3)*(x3)) + -2.5129*((x0)*(x0)*(x0)) + -0.7887*((x0)*(x0)*(x1)) + -0.9308*((x0)*(x0)*(x2)) + 0.8396*((x0)*(x0)*(x3)) + 2.2264*((x0)*(x1)*(x1)) + -2.8761*((x0)*(x1)*(x2)) + 5.4644*((x0)*(x1)*(x3)) + -1.2758*((x0)*(x2)*(x2)) + -0.5072*((x0)*(x2)*(x3)) + 1.2513*((x0)*(x3)*(x3)) + 1.0118*((x1)*(x1)*(x1)) + 3.3982*((x1)*(x1)*(x2)) + -3.18*((x1)*(x1)*(x3)) + 0.3328*((x1)*(x2)*(x2)) + -0.4559*((x1)*(x2)*(x3)) + -1.23*((x1)*(x3)*(x3)) + -0.4279*((x2)*(x2)*(x2)) + 0.6184*((x2)*(x3)*(x3)) + 1.6031*((x3)*(x3)*(x3)) + 1.1161*((x0)*(x0)*(x0)*(x0)) + -0.4904*((x0)*(x0)*(x0)*(x1)) + -0.5778*((x0)*(x0)*(x1)*(x1)) + 0.8791*((x0)*(x0)*(x1)*(x3)) + 0.862*((x1)*(x1)*(x1)*(x1)) + -0.6827*((x1)*(x2)*(x3)*(x3)) + 0.5879*((x2)*(x3)*(x3)*(x3)) + 0.3637*((x3)*(x3)*(x3)*(x3)) + -161.4295*(sin((x0))) + -103.5718*(sin((x1))) + -118.3949*(sin((x2))) + 23.4749*(sin((x3))) + -121.3485*(sin((x0)*(x0))) + -42.4576*(sin((x0)*(x1))) + -106.2406*(sin((x0)*(x2))) + -67.3342*(sin((x0)*(x3))) + 142.5271*(sin((x1)*(x1))) + 43.5748*(sin((x1)*(x2))) + 183.4953*(sin((x1)*(x3))) + 197.3063*(sin((x2)*(x2))) + -95.3066*(sin((x2)*(x3))) + -37.2968*(sin((x3)*(x3))) + 93.5651*(sin((x0)*(x0)*(x0))) + -87.4954*(sin((x0)*(x0)*(x1))) + 80.4485*(sin((x0)*(x0)*(x2))) + -138.9196*(sin((x0)*(x0)*(x3))) + -52.8786*(sin((x0)*(x1)*(x1))) + 28.8297*(sin((x0)*(x1)*(x2))) + 121.8495*(sin((x0)*(x1)*(x3))) + -214.7837*(sin((x0)*(x2)*(x2))) + -28.6206*(sin((x0)*(x2)*(x3))) + -36.4439*(sin((x0)*(x3)*(x3))) + -98.9876*(sin((x1)*(x1)*(x1))) + 100.3342*(sin((x1)*(x1)*(x2))) + 123.0186*(sin((x1)*(x1)*(x3))) + -85.7792*(sin((x1)*(x2)*(x2))) + 84.5002*(sin((x1)*(x2)*(x3))) + 50.1801*(sin((x1)*(x3)*(x3))) + 24.6718*(sin((x2)*(x2)*(x2))) + 90.78*(sin((x2)*(x2)*(x3))) + -41.7005*(sin((x2)*(x3)*(x3))) + 38.8912*(sin((x3)*(x3)*(x3))) + -43.2639*(sin((x0)*(x0)*(x0)*(x1))) + -44.231*(sin((x0)*(x0)*(x0)*(x2))) + -178.7748*(sin((x0)*(x0)*(x0)*(x3))) + 48.155*(sin((x0)*(x0)*(x1)*(x1))) + -0.5934*(sin((x0)*(x0)*(x1)*(x2))) + -42.7446*(sin((x0)*(x0)*(x1)*(x3))) + 45.5891*(sin((x0)*(x0)*(x2)*(x2))) + -184.0262*(sin((x0)*(x0)*(x2)*(x3))) + -70.3187*(sin((x0)*(x0)*(x3)*(x3))) + 34.3814*(sin((x0)*(x1)*(x1)*(x1))) + 81.077*(sin((x0)*(x1)*(x1)*(x2))) + -100.2733*(sin((x0)*(x1)*(x1)*(x3))) + -119.1282*(sin((x0)*(x1)*(x2)*(x2))) + -169.7972*(sin((x0)*(x1)*(x2)*(x3))) + -1.1829*(sin((x0)*(x1)*(x3)*(x3))) + 106.7441*(sin((x0)*(x2)*(x2)*(x2))) + -23.143*(sin((x0)*(x2)*(x2)*(x3))) + -26.5897*(sin((x0)*(x2)*(x3)*(x3))) + 32.252*(sin((x0)*(x3)*(x3)*(x3))) + -143.8076*(sin((x1)*(x1)*(x1)*(x1))) + 35.3576*(sin((x1)*(x1)*(x1)*(x2))) + -45.1745*(sin((x1)*(x1)*(x1)*(x3))) + -176.6767*(sin((x1)*(x1)*(x2)*(x2))) + 65.1481*(sin((x1)*(x1)*(x2)*(x3))) + -62.4891*(sin((x1)*(x1)*(x3)*(x3))) + 37.5239*(sin((x1)*(x2)*(x2)*(x2))) + 112.2308*(sin((x1)*(x2)*(x2)*(x3))) + 158.5796*(sin((x1)*(x2)*(x3)*(x3))) + 69.6419*(sin((x1)*(x3)*(x3)*(x3))) + 43.4266*(sin((x2)*(x2)*(x2)*(x2))) + 186.8069*(sin((x2)*(x2)*(x2)*(x3))) + -109.4988*(sin((x2)*(x2)*(x3)*(x3))) + -17.7371*(sin((x2)*(x3)*(x3)*(x3))) + -20.1993*(sin((x3)*(x3)*(x3)*(x3))) + 14.3709*(1/((x0))) + 33.9452*(1/((x1))) + -125.1699*(1/((x2))) + -16.0134*(1/((x3))) + 0.4927*(1/((x0)*(x0))) + -15.9528*(1/((x0)*(x1))) + -126.3289*(1/((x0)*(x2))) + -57.3529*(1/((x0)*(x3))) + 0.3218*(1/((x1)*(x1))) + 26.8619*(1/((x1)*(x2))) + -44.2686*(1/((x1)*(x3))) + 2.7723*(1/((x2)*(x2))) + 93.0298*(1/((x2)*(x3))) + -0.8195*(1/((x3)*(x3))) + 5.4602*(1/((x0)*(x0)*(x1))) + -3.1952*(1/((x0)*(x0)*(x2))) + 12.5889*(1/((x0)*(x0)*(x3))) + -5.9522*(1/((x0)*(x1)*(x1))) + -75.331*(1/((x0)*(x1)*(x2))) + -86.5389*(1/((x0)*(x1)*(x3))) + 5.323*(1/((x0)*(x2)*(x2))) + 94.2167*(1/((x0)*(x2)*(x3))) + 5.0416*(1/((x0)*(x3)*(x3))) + -6.3886*(1/((x1)*(x1)*(x2))) + 0.9066*(1/((x1)*(x1)*(x3))) + 3.7473*(1/((x1)*(x2)*(x2))) + -9.1521*(1/((x1)*(x2)*(x3))) + -1.1874*(1/((x1)*(x3)*(x3))) + -2.9834*(1/((x2)*(x2)*(x3))) + 1.2524*(1/((x2)*(x3)*(x3))) + -0.3457*(1/((x0)*(x0)*(x1)*(x1))) + -0.935*(1/((x0)*(x0)*(x1)*(x2))) + -28.7527*(1/((x0)*(x0)*(x1)*(x3))) + 0.4255*(1/((x0)*(x0)*(x2)*(x2))) + -36.1174*(1/((x0)*(x0)*(x2)*(x3))) + 1.3659*(1/((x0)*(x0)*(x3)*(x3))) + 31.0912*(1/((x0)*(x1)*(x1)*(x2))) + 1.5663*(1/((x0)*(x1)*(x1)*(x3))) + -5.116*(1/((x0)*(x1)*(x2)*(x2))) + 11.1347*(1/((x0)*(x1)*(x2)*(x3))) + 1.2688*(1/((x0)*(x1)*(x3)*(x3))) + -6.7378*(1/((x0)*(x2)*(x2)*(x3))) + 7.0788*(1/((x0)*(x2)*(x3)*(x3))) + -0.8758*(1/((x1)*(x2)*(x2)*(x3))) + -2.4722*(1/((x1)*(x2)*(x3)*(x3))) + 822.3097	-0.3871525051

			0.4	-197.105*((x0)) + -52.2205*((x1)) + -44.1226*((x2)) + -82.0357*((x3)) + -4.2863*((x0)*(x0)) + -4.9829*((x0)*(x1)) + -29.0118*((x0)*(x2)) + 27.8047*((x0)*(x3)) + 20.1202*((x1)*(x1)) + 14.2769*((x1)*(x2)) + 6.1987*((x1)*(x3)) + 40.6056*((x2)*(x2)) + -13.1157*((x2)*(x3)) + 22.5039*((x3)*(x3)) + 0.9282*((x0)*(x0)*(x1)) + -4.0615*((x0)*(x0)*(x2)) + -2.8809*((x0)*(x0)*(x3)) + 2.8474*((x0)*(x1)*(x1)) + 0.3499*((x0)*(x1)*(x2)) + -1.3219*((x0)*(x1)*(x3)) + 1.8326*((x0)*(x2)*(x2)) + 1.0726*((x0)*(x2)*(x3)) + 0.5424*((x0)*(x3)*(x3)) + -1.6196*((x1)*(x1)*(x1)) + 3.0407*((x1)*(x1)*(x2)) + 0.5983*((x1)*(x1)*(x3)) + 4.019*((x1)*(x2)*(x2)) + -1.9817*((x1)*(x2)*(x3)) + -1.0902*((x1)*(x3)*(x3)) + -2.0166*((x2)*(x2)*(x2)) + 1.0794*((x2)*(x2)*(x3)) + 5.6376*((x2)*(x3)*(x3)) + 1.5588*((x3)*(x3)*(x3)) + 2.8197*((x0)*(x0)*(x0)*(x1)) + 1.1922*((x0)*(x0)*(x0)*(x3)) + -1.3727*((x0)*(x0)*(x1)*(x1)) + -0.5306*((x0)*(x0)*(x1)*(x2)) + 1.3293*((x0)*(x0)*(x1)*(x3)) + 3.5908*((x0)*(x0)*(x2)*(x3)) + 1.3162*((x0)*(x0)*(x3)*(x3)) + -2.1479*((x0)*(x1)*(x1)*(x1)) + 3.7803*((x0)*(x1)*(x1)*(x2)) + -2.345*((x0)*(x1)*(x1)*(x3)) + -1.1473*((x0)*(x1)*(x2)*(x2)) + 1.2681*((x0)*(x1)*(x2)*(x3)) + -0.9758*((x0)*(x2)*(x2)*(x2)) + 0.4325*((x0)*(x2)*(x2)*(x3)) + -1.4284*((x0)*(x2)*(x3)*(x3)) + -0.5871*((x0)*(x3)*(x3)*(x3)) + 0.8894*((x1)*(x1)*(x1)*(x1)) + 2.2802*((x1)*(x1)*(x1)*(x2)) + 0.6423*((x1)*(x1)*(x1)*(x3)) + -1.1342*((x1)*(x1)*(x3)*(x3)) + -1.208*((x1)*(x2)*(x2)*(x2)) + -0.4518*((x1)*(x2)*(x2)*(x3)) + -1.9344*((x1)*(x2)*(x3)*(x3)) + -1.3176*((x1)*(x3)*(x3)*(x3)) + -0.5641*((x2)*(x2)*(x2)*(x3)) + -0.3943*((x2)*(x2)*(x3)*(x3)) + -1.31*((x2)*(x3)*(x3)*(x3)) + -10.2744*(sin((x0))) + -32.4991*(sin((x1))) + -43.74*(sin((x2))) + -54.9394*(sin((x3))) + -104.2983*(sin((x0)*(x0))) + 21.3623*(sin((x0)*(x1))) + 42.3566*(sin((x0)*(x2))) + 16.4419*(sin((x0)*(x3))) + -29.5615*(sin((x1)*(x1))) + -57.5718*(sin((x1)*(x2))) + -28.4307*(sin((x1)*(x3))) + 8.7861*(sin((x2)*(x2))) + -34.7178*(sin((x2)*(x3))) + -60.2767*(sin((x3)*(x3))) + 54.0708*(sin((x0)*(x0)*(x0))) + -3.4012*(sin((x0)*(x0)*(x1))) + 47.5021*(sin((x0)*(x0)*(x2))) + -65.1088*(sin((x0)*(x0)*(x3))) + -136.9926*(sin((x0)*(x1)*(x1))) + 125.0979*(sin((x0)*(x1)*(x2))) + 73.0331*(sin((x0)*(x1)*(x3))) + -21.8179*(sin((x0)*(x2)*(x2))) + -2.6382*(sin((x0)*(x2)*(x3))) + -118.0216*(sin((x0)*(x3)*(x3))) + -77.9572*(sin((x1)*(x1)*(x1))) + 5.9804*(sin((x1)*(x1)*(x2))) + 36.6593*(sin((x1)*(x1)*(x3))) + 105.0135*(sin((x1)*(x2)*(x2))) + 97.6453*(sin((x1)*(x2)*(x3))) + -92.4172*(sin((x1)*(x3)*(x3))) + -36.1231*(sin((x2)*(x2)*(x2))) + -119.4255*(sin((x2)*(x2)*(x3))) + -22.5182*(sin((x2)*(x3)*(x3))) + 12.1006*(sin((x3)*(x3)*(x3))) + 79.7745*(sin((x0)*(x0)*(x0)*(x0))) + 129.1322*(sin((x0)*(x0)*(x0)*(x1))) + 67.407*(sin((x0)*(x0)*(x0)*(x2))) + 55.7395*(sin((x0)*(x0)*(x0)*(x3))) + 95.7765*(sin((x0)*(x0)*(x1)*(x1))) + -30.7524*(sin((x0)*(x0)*(x1)*(x2))) + -64.3279*(sin((x0)*(x0)*(x1)*(x3))) + -2.0994*(sin((x0)*(x0)*(x2)*(x2))) + 10.8964*(sin((x0)*(x0)*(x2)*(x3))) + -73.0431*(sin((x0)*(x0)*(x3)*(x3))) + -16.4428*(sin((x0)*(x1)*(x1)*(x1))) + 146.0164*(sin((x0)*(x1)*(x1)*(x2))) + -27.6292*(sin((x0)*(x1)*(x1)*(x3))) + 49.3131*(sin((x0)*(x1)*(x2)*(x2))) + 72.9992*(sin((x0)*(x1)*(x2)*(x3))) + 238.1191*(sin((x0)*(x1)*(x3)*(x3))) + 79.9625*(sin((x0)*(x2)*(x2)*(x2))) + 12.4974*(sin((x0)*(x2)*(x2)*(x3))) + 59.2076*(sin((x0)*(x2)*(x3)*(x3))) + -3.1742*(sin((x0)*(x3)*(x3)*(x3))) + -16.0643*(sin((x1)*(x1)*(x1)*(x1))) + 53.3232*(sin((x1)*(x1)*(x1)*(x2))) + -65.9077*(sin((x1)*(x1)*(x1)*(x3))) + -61.1392*(sin((x1)*(x1)*(x2)*(x2))) + 27.0383*(sin((x1)*(x1)*(x2)*(x3))) + -20.3815*(sin((x1)*(x1)*(x3)*(x3))) + -11.6614*(sin((x1)*(x2)*(x2)*(x2))) + -123.5791*(sin((x1)*(x2)*(x2)*(x3))) + -11.8687*(sin((x1)*(x2)*(x3)*(x3))) + 3.1702*(sin((x1)*(x3)*(x3)*(x3))) + 92.8412*(sin((x2)*(x2)*(x2)*(x2))) + 2.2977*(sin((x2)*(x2)*(x2)*(x3))) + -25.8252*(sin((x2)*(x2)*(x3)*(x3))) + -27.4612*(sin((x2)*(x3)*(x3)*(x3))) + -56.4972*(sin((x3)*(x3)*(x3)*(x3))) + -191.6131*(1/((x0))) + -60.9338*(1/((x1))) + 50.2234*(1/((x2))) + -34.9057*(1/((x3))) + 0.4862*(1/((x0)*(x0))) + 15.3567*(1/((x0)*(x1))) + 40.2184*(1/((x0)*(x2))) + -52.7246*(1/((x0)*(x3))) + 37.2659*(1/((x1)*(x2))) + -17.1819*(1/((x1)*(x3))) + 12.3772*(1/((x2)*(x3))) + -2.5068*(1/((x0)*(x0)*(x1))) + 0.48*(1/((x0)*(x0)*(x2))) + 3.3307*(1/((x0)*(x0)*(x3))) + 0.4155*(1/((x0)*(x1)*(x1))) + 27.4087*(1/((x0)*(x1)*(x2))) + -51.6177*(1/((x0)*(x1)*(x3))) + -20.0442*(1/((x0)*(x2)*(x3))) + -1.1482*(1/((x1)*(x1)*(x2))) + 0.3256*(1/((x1)*(x1)*(x3))) + 0.7065*(1/((x1)*(x2)*(x2))) + -8.2517*(1/((x1)*(x2)*(x3))) + 0.8426*(1/((x2)*(x2)*(x3))) + -1629.129	-6.880727857

			0.6	1.3223*((x0)) + -16.6717*((x1)) + -17.9198*((x2)) + 1.9294*((x3)) + -4.1921*((x0)*(x0)) + 1.9008*((x0)*(x1)) + -0.5*((x0)*(x3)) + -3.4241*((x1)*(x2)) + -1.3811*((x2)*(x2)) + 2.9112*((x2)*(x3)) + 0.5754*((x3)*(x3)) + 0.4176*((x0)*(x0)*(x1)) + -0.3187*((x0)*(x3)*(x3)) + -2.1822*(sin((x0))) + -4.0063*(sin((x1))) + -3.0158*(sin((x2))) + 8.7177*(sin((x3))) + -1.5691*(sin((x0)*(x0))) + 8.9775*(sin((x0)*(x1))) + 10.8297*(sin((x0)*(x2))) + 14.2779*(sin((x0)*(x3))) + 18.6269*(sin((x1)*(x1))) + 3.9146*(sin((x1)*(x2))) + -6.3857*(sin((x1)*(x3))) + 8.3455*(sin((x2)*(x2))) + 7.357*(sin((x2)*(x3))) + 9.3077*(sin((x3)*(x3))) + -2.0419*(sin((x0)*(x0)*(x0))) + 8.1187*(sin((x0)*(x0)*(x1))) + -8.1447*(sin((x0)*(x0)*(x2))) + -3.2382*(sin((x0)*(x0)*(x3))) + -12.4409*(sin((x0)*(x1)*(x1))) + 20.8126*(sin((x0)*(x1)*(x2))) + 1.9522*(sin((x0)*(x1)*(x3))) + -17.3792*(sin((x0)*(x2)*(x2))) + 32.8967*(sin((x0)*(x2)*(x3))) + 19.6441*(sin((x0)*(x3)*(x3))) + -20.776*(sin((x1)*(x1)*(x1))) + 8.5627*(sin((x1)*(x1)*(x2))) + 17.6244*(sin((x1)*(x1)*(x3))) + -19.6116*(sin((x1)*(x2)*(x2))) + -29.4305*(sin((x1)*(x2)*(x3))) + -10.0829*(sin((x1)*(x3)*(x3))) + 19.0875*(sin((x2)*(x2)*(x2))) + 1.5188*(sin((x2)*(x2)*(x3))) + 1.0868*(sin((x2)*(x3)*(x3))) + -13.1311*(sin((x3)*(x3)*(x3))) + -6.2346*(sin((x0)*(x0)*(x0)*(x0))) + 7.2331*(sin((x0)*(x0)*(x0)*(x1))) + -8.6583*(sin((x0)*(x0)*(x0)*(x2))) + -12.3737*(sin((x0)*(x0)*(x0)*(x3))) + 5.7341*(sin((x0)*(x0)*(x1)*(x1))) + -11.1643*(sin((x0)*(x0)*(x1)*(x2))) + -11.1822*(sin((x0)*(x0)*(x1)*(x3))) + -14.5407*(sin((x0)*(x0)*(x2)*(x2))) + -15.9698*(sin((x0)*(x0)*(x2)*(x3))) + -5.3532*(sin((x0)*(x0)*(x3)*(x3))) + -29.2225*(sin((x0)*(x1)*(x1)*(x1))) + -18.5602*(sin((x0)*(x1)*(x1)*(x2))) + -6.434*(sin((x0)*(x1)*(x1)*(x3))) + -4.5743*(sin((x0)*(x1)*(x2)*(x2))) + 2.3455*(sin((x0)*(x1)*(x2)*(x3))) + 3.2047*(sin((x0)*(x1)*(x3)*(x3))) + 3.3037*(sin((x0)*(x2)*(x2)*(x2))) + -29.27*(sin((x0)*(x2)*(x2)*(x3))) + 5.5673*(sin((x0)*(x2)*(x3)*(x3))) + -10.2152*(sin((x0)*(x3)*(x3)*(x3))) + 3.7783*(sin((x1)*(x1)*(x1)*(x1))) + 7.5488*(sin((x1)*(x1)*(x1)*(x2))) + -8.5897*(sin((x1)*(x1)*(x1)*(x3))) + -5.1083*(sin((x1)*(x1)*(x2)*(x2))) + -28.4791*(sin((x1)*(x1)*(x2)*(x3))) + -14.4065*(sin((x1)*(x1)*(x3)*(x3))) + 1.1669*(sin((x1)*(x2)*(x2)*(x2))) + 23.7657*(sin((x1)*(x2)*(x2)*(x3))) + 13.4939*(sin((x1)*(x2)*(x3)*(x3))) + -8.3633*(sin((x1)*(x3)*(x3)*(x3))) + -25.1084*(sin((x2)*(x2)*(x2)*(x2))) + 21.9947*(sin((x2)*(x2)*(x2)*(x3))) + 4.7998*(sin((x2)*(x2)*(x3)*(x3))) + 10.2522*(sin((x2)*(x3)*(x3)*(x3))) + -11.1763*(sin((x3)*(x3)*(x3)*(x3))) + 2.9163*(1/((x1))) + -3.7235*(1/((x2))) + -3.0367*(1/((x3))) + -9.6314*(1/((x0)*(x1))) + 4.0831*(1/((x0)*(x2))) + -2.1703*(1/((x0)*(x3))) + 1.5339*(1/((x1)*(x2))) + -3.0507*(1/((x1)*(x3))) + 4.0321*(1/((x2)*(x3))) + 0.8729*(1/((x0)*(x1)*(x2))) + -3.2546*(1/((x0)*(x1)*(x3))) + -8.3521*(1/((x0)*(x2)*(x3))) + 0.6459*(1/((x1)*(x1)*(x2))) + -0.9419*(1/((x1)*(x2)*(x2))) + 3.4916*(1/((x1)*(x2)*(x3))) + 201.9801	-0.332290674
13	T = 2pi/w	2pi/x0	0	6.2832*(1/((x0))) + 0.0	1
			0.2	0.5787*((x0)) + 0.4887*(1/((x0))) + -3.1407	0.002906234534
			0.4	0.4304*((x0)) + 1.1704*(1/((x0))) + -2.623	0.003101559591
			0.6	0.5425*((x0)) + -0.3904*(1/((x0))) + 0.089	0.0004904541162
14	F_c = m v^2/R	x0 * x1^2 / x2	0	-1.0481*((x0)) + -0.8351*((x1)) + 17.1809*((x2)) + -3.249*((x0)*(x0)) + -10.1909*((x0)*(x1)) + 0.3165*((x0)*(x2)) + 4.1446*((x1)*(x1)) + 54.8882*((x1)*(x2)) + 18.5894*((x2)*(x2)) + -0.3705*((x1)*(x1)*(x1)*(x2)) + -0.3774*((x1)*(x2)*(x2)*(x2)) + 85.4161*(1/((x0))) + -2.1229*(1/((x1))) + -129.7458*(1/((x2))) + -0.5499*(1/((x0)*(x0))) + 122.3791*(1/((x0)*(x1))) + 231.7337*(1/((x0)*(x2))) + -24.0946*(1/((x1)*(x2))) + 0.357*(1/((x2)*(x2))) + -1.8106*(1/((x0)*(x0)*(x1))) + -7.6025*(1/((x0)*(x0)*(x2))) + 0.6462*(1/((x0)*(x1)*(x1))) + -3.143*(1/((x0)*(x1)*(x2))) + 1.6463*(1/((x0)*(x2)*(x2))) + 0.4433*(1/((x1)*(x1)*(x2))) + -1.1811*(1/((x1)*(x2)*(x2))) + 0.5503*(1/((x0)*(x0)*(x1)*(x1))) + 0.3178*(1/((x0)*(x0)*(x1)*(x2))) + -0.6152*(1/((x0)*(x0)*(x2)*(x2))) + -1.457*(1/((x0)*(x1)*(x1)*(x2))) + -0.36*(1/((x0)*(x1)*(x2)*(x2))) + -285.0321	0.1479885605

			0.2	-1231.1588*((x0)) + 654.5669*((x1)) + 1146.1351*((x2)) + -8.1439*((x0)*(x0)) + 92.0103*((x0)*(x1)) + -49.6698*((x0)*(x2)) + -13.5505*((x1)*(x1)) + 8.7376*((x1)*(x2)) + -47.3085*((x2)*(x2)) + 9.6868*((x0)*(x0)*(x0)) + -6.7387*((x0)*(x0)*(x1)) + -5.3053*((x0)*(x0)*(x2)) + 2.4492*((x0)*(x1)*(x1)) + 0.4334*((x0)*(x1)*(x2)) + 9.3899*((x0)*(x2)*(x2)) + -7.1691*((x1)*(x1)*(x1)) + -6.1545*((x1)*(x1)*(x2)) + 0.409*((x1)*(x2)*(x2)) + -11.0989*((x2)*(x2)*(x2)) + 0.4467*((x0)*(x0)*(x0)*(x2)) + 0.4534*((x0)*(x0)*(x1)*(x1)) + -0.5255*((x0)*(x0)*(x2)*(x2)) + -0.6267*((x0)*(x1)*(x1)*(x1)) + -0.9578*((x0)*(x1)*(x2)*(x2)) + 0.3351*((x0)*(x2)*(x2)*(x2)) + 0.6877*((x2)*(x2)*(x2)*(x2)) + 327.5463*(1/((x0))) + -155.7115*(1/((x1))) + 1368.7326*(1/((x2))) + 369.4365*(1/((x0)*(x1))) + 469.0132*(1/((x0)*(x2))) + -2.8882*(1/((x1)*(x1))) + -517.2043*(1/((x1)*(x2))) + -3.1198*(1/((x2)*(x2))) + 2.7982*(1/((x0)*(x0)*(x1))) + 5.8052*(1/((x0)*(x0)*(x2))) + 2.2074*(1/((x0)*(x1)*(x1))) + 232.1327*(1/((x0)*(x1)*(x2))) + 7.4982*(1/((x0)*(x2)*(x2))) + 12.4282*(1/((x1)*(x2)*(x2))) + -0.3262*(1/((x0)*(x0)*(x1)*(x1))) + 0.3228*(1/((x0)*(x0)*(x2)*(x2))) + 451.9674	-0.0322427766
			0.4	42.5489*((x0)) + -38.8095*((x1)) + -15.9248*((x2)) + -21.3768*((x0)*(x0)) + -7.1095*((x0)*(x1)) + -8.9963*((x0)*(x2)) + 28.1526*((x1)*(x1)) + -2.6281*((x1)*(x2)) + 10.7258*((x2)*(x2)) + -0.4276*((x0)*(x0)*(x0)) + 0.4076*((x1)*(x1)*(x1)) + -66.6971*(1/((x0))) + 49.3689*(1/((x1))) + 23.4908*(1/((x2))) + 18.1639*(1/((x0)*(x1))) + 30.0728*(1/((x0)*(x2))) + 34.7021*(1/((x1)*(x2))) + -0.8257*(1/((x0)*(x0)*(x1))) + -0.5469*(1/((x0)*(x0)*(x2))) + -110.3462*(1/((x0)*(x1)*(x2))) + -0.3681*(1/((x0)*(x1)*(x1)*(x2))) + -1.0827*(1/((x0)*(x1)*(x2)*(x2))) + -260.7858	-0.2591199204
			0.6	-19.5058*((x0)) + -21.3681*((x1)) + -16.2966*((x2)) + 1.8414*((x0)*(x0)) + -1.8478*((x0)*(x1)) + 1.5433*((x0)*(x2)) + 1.121*((x1)*(x2)) + 4.7311*((x2)*(x2)) + -0.8094*((x0)*(x0)*(x2)) + 0.3607*((x0)*(x1)*(x1)) + 0.4269*((x0)*(x2)*(x2)) + -0.6615*((x1)*(x1)*(x2)) + -1.2494*((x2)*(x2)*(x2)) + -4.622*(1/((x0))) + 1.3861*(1/((x1))) + -5.1492*(1/((x2))) + 2.2954*(1/((x0)*(x0))) + 9.0738*(1/((x0)*(x1))) + -2.3348*(1/((x0)*(x2))) + 0.4282*(1/((x1)*(x1))) + 4.8596*(1/((x1)*(x2))) + -1.1337*(1/((x0)*(x0)*(x1))) + -3.3756*(1/((x0)*(x0)*(x2))) + 0.4608*(1/((x0)*(x1)*(x1))) + 0.4584*(1/((x0)*(x1)*(x2))) + -0.6677*(1/((x0)*(x2)*(x2))) + 1.1991*(1/((x1)*(x1)*(x2))) + 1.742*(1/((x1)*(x2)*(x2))) + -0.7207*(1/((x0)*(x1)*(x1)*(x2))) + 1.1579*(1/((x0)*(x1)*(x2)*(x2))) + -0.6932*(1/((x1)*(x1)*(x2)*(x2))) + -170.9459	-0.2712867235
15	F = mgcos(theta)	x0*x1*cos(x2)	0	-0.7403*(cos((x1))) + -0.8188*(cos((x0)*(x1))) + -0.5346*(cos((x0)*(x2))) + 1.0685*(cos((x1)*(x1))) + 0.8773*(cos((x0)*(x0)*(x2))) + 0.3654*(cos((x0)*(x1)*(x1))) + -0.4945*(cos((x1)*(x1)*(x1))) + 0.3784*(cos((x1)*(x2)*(x2))) + -0.4403*(cos((x2)*(x2)*(x2))) + 0.3767*(cos((x0)*(x0)*(x1)*(x1))) + 0.5725*(cos((x0)*(x1)*(x1)*(x1))) + 0.3554*(cos((x1)*(x1)*(x1)*(x1))) + -0.7106*(cos((x1)*(x2)*(x2)*(x2))) + -1.3525	0.07919453937
			0.2	0.7734*(cos((x2))) + -0.4515*(cos((x0)*(x0))) + -0.5481*(cos((x0)*(x1))) + -0.5718*(cos((x0)*(x2))) + -0.3617*(cos((x2)*(x2))) + -0.3279*(cos((x0)*(x0)*(x2))) + 0.3325*(cos((x0)*(x1)*(x2))) + 0.343*(cos((x1)*(x2)*(x2))) + 0.3764*(cos((x2)*(x2)*(x2))) + 0.7019*(cos((x0)*(x0)*(x0)*(x0))) + 0.7156*(cos((x0)*(x0)*(x0)*(x1))) + -0.4527*(cos((x0)*(x0)*(x0)*(x2))) + 0.6427*(cos((x0)*(x0)*(x1)*(x2))) + -0.6288*(cos((x0)*(x0)*(x2)*(x2))) + 0.4215*(cos((x0)*(x1)*(x2)*(x2))) + 0.3183*(cos((x0)*(x2)*(x2)*(x2))) + -0.3223*(cos((x1)*(x1)*(x2)*(x2))) + 0.3816*(cos((x1)*(x2)*(x2)*(x2))) + -0.4273*(cos((x2)*(x2)*(x2)*(x2))) + -0.2724	0.06815280796
			0.4	0.7806*(cos((x2))) + -0.4165*(cos((x0)*(x0))) + -0.8098*(cos((x0)*(x0)*(x0))) + 0.4363*(cos((x0)*(x0)*(x1))) + 0.4554*(cos((x0)*(x1)*(x1))) + -0.807*(cos((x1)*(x1)*(x1))) + 0.512*(cos((x0)*(x1)*(x1)*(x2))) + 0.6052*(cos((x0)*(x1)*(x2)*(x2))) + -0.3385*(cos((x0)*(x2)*(x2)*(x2))) + -0.2196	0.05859211564
			0.6	-0.3909*(cos((x1)*(x2))) + 0.5089*(cos((x0)*(x0)*(x2))) + -0.5326*(cos((x0)*(x1)*(x2))) + 0.3739*(cos((x1)*(x1)*(x1))) + 0.6875*(cos((x2)*(x2)*(x2))) + -0.458*(cos((x0)*(x0)*(x0)*(x2))) + 0.6023*(cos((x0)*(x0)*(x1)*(x2))) + 0.3484*(cos((x0)*(x0)*(x2)*(x2))) + -0.3222*(cos((x0)*(x1)*(x2)*(x2))) + -0.4295*(cos((x1)*(x1)*(x1)*(x1))) + -0.4141*(cos((x1)*(x1)*(x2)*(x2))) + 0.3396*(cos((x2)*(x2)*(x2)*(x2))) + 0.1961	0.08060896995

16	Stress strain relationship	x0 / x1 + (x0 / x2)**x3	0	-4.5661*((x0)) + 5.9707*((x1)) + -1.1108*((x2)) + 2.8429*((x0)*(x0)) + -11.7622*((x0)*(x1)) + -1.2376*((x0)*(x2)) + -4.1219*((x0)*(x3)) + 12.3554*((x1)*(x1)) + -1.6275*((x2)*(x2)) + 4.073*((x2)*(x3)) + 3.8996*((x0)*(x0)*(x0)) + -2.3619*((x0)*(x0)*(x2)) + -5.9087*((x0)*(x0)*(x3)) + -10.7056*((x0)*(x1)*(x1)) + -0.7773*((x0)*(x2)*(x2)) + 5.7879*((x0)*(x2)*(x3)) + 11.1443*((x1)*(x1)*(x1)) + -0.7151*((x2)*(x2)*(x2)) + 2.1175*((x0)*(x0)*(x0)*(x0)) + -1.7372*((x0)*(x0)*(x0)*(x2)) + -3.7068*((x0)*(x0)*(x0)*(x3)) + 4.2208*((x0)*(x0)*(x2)*(x3)) + 1.1677*((x0)*(x0)*(x3)*(x3)) + -3.6084*((x0)*(x1)*(x1)*(x1)) + -2.0152*((x0)*(x2)*(x3)*(x3)) + 3.7194*((x1)*(x1)*(x1)*(x1)) + -0.3656*((x2)*(x2)*(x2)*(x3)) + 0.9347*((x2)*(x2)*(x3)*(x3)) + 2.0602	0.9999956588
			0.2	2.6512*((x0)) + 1.3719*((x1)) + 1.6056*((x2)) + 1.8566*((x3)) + 1.8067*((x0)*(x0)) + 1.5473*((x0)*(x1)) + 4.7744*((x0)*(x2)) + 2.1617*((x0)*(x3)) + 1.3656*((x1)*(x2)) + 2.3586*((x1)*(x3)) + 0.9852*((x2)*(x2)) + 1.574*((x3)*(x3)) + 1.0932*((x0)*(x0)*(x1)) + 3.4633*((x0)*(x0)*(x2)) + 0.623*((x0)*(x0)*(x3)) + -0.7607*((x0)*(x1)*(x1)) + 3.609*((x0)*(x1)*(x2)) + 1.644*((x0)*(x1)*(x3)) + 0.5078*((x0)*(x2)*(x2)) + 1.3043*((x0)*(x2)*(x3)) + 1.0654*((x0)*(x3)*(x3)) + 0.8851*((x1)*(x1)*(x3)) + 1.3125*((x1)*(x3)*(x3)) + 0.624*((x2)*(x2)*(x2)) + -0.7719*((x2)*(x2)*(x3)) + 0.5446*((x3)*(x3)*(x3)) + 0.5657*((x0)*(x0)*(x0)*(x2)) + -0.4091*((x0)*(x0)*(x0)*(x3)) + 1.3993*((x0)*(x0)*(x1)*(x2)) + 0.801*((x0)*(x0)*(x1)*(x3)) + 0.8558*((x0)*(x0)*(x2)*(x3)) + -1.1484*((x0)*(x1)*(x1)*(x3)) + 0.9997*((x0)*(x1)*(x2)*(x2)) + 0.371*((x0)*(x1)*(x2)*(x3)) + 1.3508*((x0)*(x1)*(x3)*(x3)) + 0.4294*((x1)*(x1)*(x1)*(x3)) + -0.7423*((x1)*(x2)*(x2)*(x3)) + 3.2861	0.2140824055
			0.4	0.4943*((x3)) + -0.3772*((x0)*(x2)) + 0.5866*((x0)*(x3)) + 0.3748*((x1)*(x2)) + 0.6418*((x1)*(x3)) + 0.5176*((x2)*(x2)) + 0.3761*((x2)*(x3)) + 0.3243*((x3)*(x3)) + -0.4128*((x0)*(x1)*(x2)) + 0.5557*((x0)*(x1)*(x3)) + 0.3694*((x1)*(x2)*(x2)) + 0.4873*((x1)*(x2)*(x3)) + 0.3479*((x1)*(x3)*(x3)) + 0.3399*((x2)*(x2)*(x2)) + 0.3999*((x2)*(x3)*(x3)) + 2.1756	0.04160926978
			0.6	2.0137	0.01661764946

Noise Level	Average R^2
0.0	0.8135730501
0.2	0.494231259
0.4	-0.1077119099
0.6	0.1557428572