Comparison results of different symbolic regression methods

Content

EFS

Remark:
efs=Evolutionary Feature Synthesis
EFS can converge in 1 minute for all test cases

efs.jar

artifical_plus.csv

features X1,X2,X3,X4,(* (+ X4 X1) X1),X1,(- X0 X0),(+ X4 X1),(* X3 (mydivide X1 X3)),(mydivide (+ X4 X1) X3),(- (sin X2) (sin X1)),(- (sin X1) X1),(+ (+ X4 X1) X1),(square (- (sin X1) X1)),(- (sin X1) (cos X3)),(cube (- (sin X1) X1)),

model
102.52,112.2
100.55995586025419
  + 0.017650136206987282 * X1
  + 0.0036553571172496524 * (* (+ X4 X1) X1)
  + 0.022312405431059737 * X1
  + 1.3719529547547789E-7 * (* X3 (mydivide X1 X3))
  + 0.04428576634901983 * (mydivide (+ X4 X1) X3)
  + 0.9797377025624736 * (- (sin X2) (sin X1))
  + -0.4266300600741692 * (- (sin X1) X1)
  + 0.9640714261976043 * (+ (+ X4 X1) X1)
  + 0.031004925173474508 * (square (- (sin X1) X1))
  + -0.17851331967613354 * (- (sin X1) (cos X3))
  + -0.004768748264297034 * (cube (- (sin X1) X1))

BEST MSE IS: 5.488359831081745E-4

artifical_plus2.csv

X1,X2,X3,X4,(mylog (exp (- X2 X4))),(mylog (exp (- X1 X2))),(mylog (exp (- X1 X2))),(cube (* (cos X3) X1)),(exp (- X1 X2)),(square (- (sin X2) X2)),(cos (sin X2)),(cos (* (sin X4) X4)),(sin X1),(exp (* (cos X3) X1)),(- (sin X2) X2),(* (cos X3) X1),

-58.674,25.603
6.950591643811229
  + -6.2148791944573105 * (mylog (exp (- X2 X4)))
  + 3.3420449336691 * (mylog (exp (- X1 X2)))
  + 6.535891621610734E-5 * (mylog (exp (- X1 X2)))

+ -0.07690428305643965 * (cube (* (cos X3) X1))
+ 0.013104185552987537 * (exp (- X1 X2))
+ -3.9255993741793844 * (square (- (sin X2) X2))
+ -6.908975516402892 * (cos (sin X2))
+ 1.443374788706805 * (cos (* (sin X4) X4))
+ -1.9324941289706499 * (sin X1)
+ -0.1502562542580562 * (exp (* (cos X3) X1))
+ 2.33070174729363 * (- (sin X2) X2)
+ -0.380384707342273 * (* (cos X3) X1)

BEST MSE IS: 0.060118209286092655

artifical_times.csv

X1,X2,X3,X4,(* (* X3 X1) X2),(sin (exp (* X1 X2))),(cube (exp (+ X1 X4))),(cos (cube (sin X2))),(mydivide X1 (sin X2)),(exp (* X1 X2)),(* (+ X1 X4) X1),(* (* X1 X2) X4),(cos (cube (- X4 X3))),(* (- X4 X3) X1),(sin X3),(cos (mydivide X1 (sin X2))),

100.07,121.55
105.12001968215547
+ -2.688632663840392 * X1
+ 0.926481347149887 * X2
+ 0.4373241436239971 * X3
+ -1.4105861695888247 * X4
+ 0.27840211597929515 * (* (* X3 X1) X2)
+ 0.09705348338642723 * (sin (exp (* X1 X2)))
+ 5.995419124000901E-9 * (cube (exp (+ X1 X4)))
+ -3.7359990142388284 * (cos (cube (sin X2)))
+ -0.16810459229992672 * (mydivide X1 (sin X2))
+ -3.821382244434078E-4 * (exp (* X1 X2))
+ 1.244475571795417 * (* (+ X1 X4) X1)
+ -0.5249683917389024 * (* (* X1 X2) X4)
+ -0.2648005761747757 * (cos (cube (- X4 X3)))
+ 1.4265200451610192 * (* (- X4 X3) X1)
+ -0.7412380194760737 * (sin X3)
+ -0.5535735355763786 * (cos (mydivide X1 (sin X2)))

BEST MSE IS: 0.4333283981977147

Cd_wedge.csv

X1,X2,X3,X4,X5,(mydivide X3 X1),(cos (mydivide X3 X4)),(cos (* X4 X3)),(* (sin X4) X5),(* X3 (+ X4

X5)),(mydivide X3 X4),(cube (* X3 X1)),(sin (mylog (* X4 X3))),(sin (cube X4)),(mylog (* X4 X3)),(* X3 X1),(cube X1),(* X4 X3),(mydivide X1 (cube X4)),(- X3 (* X4 X3)),

-0.026123,0.52245
5241.530151973486
  + 0.8883894949790235 * X1
  + -0.020815310308269334 * X2
  + 0.0390293002176287 * X3
  + 3.2163149013525493E-6 * X4
  + 2.1360147915438648E-5 * X5
  + -1.2032084794585817E-5 * (mydivide X3 X1)
  + -5239.5281741743365 * (cos (mydivide X3 X4))
  + 7.710877064058126E-5 * (cos (* X4 X3))
  + 4.43859148943097E-6 * (* (sin X4) X5)
  + -2.902293796065195E-6 * (* X3 (+ X4 X5))
  + -40.50474406335109 * (mydivide X3 X4)
  + 0.017334184770760597 * (cube (* X3 X1))
  + -0.17355954453337216 * (sin (mylog (* X4 X3)))
  + -0.006616561718920441 * (sin (cube X4))
  + -0.23384550450797092 * (mylog (* X4 X3))
  + -0.16269244533199922 * (* X3 X1)
  + -1.895679286113042 * (cube X1)
  + 3.1600796123525943E-7 * (* X4 X3)
  + 7.130469000005459E8 * (mydivide X1 (cube X4))
  + -6.134168125530679E-6 * (- X3 (* X4 X3))

BEST MSE IS: 3.408355395774077E-4

Heatflux_flatplate.csv

X1,X2,X1,X1,(* X2 X1),X1,(* X1 (cos X2)),(mydivide X1 X2),

3.9684E-4,0.012392
8.075918523422826E-5
  + 0.02827779834776515 * X1
  + -9.283086026444351E-9 * X2
  + 1.2976313540933493E-6 * X1
  + 3.266004648603294E-11 * X1
  + -1.1351857673131511E-6 * (* X2 X1)
  + 6.567821239593858E-4 * X1
  + -5.971388646123781E-4 * (* X1 (cos X2))
  + 43.62414124233322 * (mydivide X1 X2)

Progress Stalled, exiting. BEST MSE IS: 2.5466379029676866E-9

Heatflux_sphere.csv

X1,X2,X3,X4,X5,(sin (square (- X3 X1))),(cos (* X1 X3)),(* X5 (* X1 X2)),(sin (cube (* X1 X3))),(cube (* X1 X3)),(- X3 (mydivide X4 X5)),(cos (square (- X3 X1))),(square (- X3 X1)),(cube (* X1 X2)),(* X1 X3),(mydivide (mydivide X4 X3) X4),(* (* X1 X3) X2),(cos (cube (* X1 X3))),(* X1 X2),(mydivide X5 (* X3 X2)),

361.69,57291.0
-2765.3870873867054
+ -18.432634875194108 * X1
+ -1.895962834548321E7 * X2
+ 310786.9587430987 * X3
+ -1.1162212500659648E-5 * X4
+ -0.001331360900516985 * X5
+ -200.97042722738058 * (sin (square (- X3 X1)))
+ -457.7657708571279 * (cos (* X1 X3))
+ 0.004073272540864728 * (* X5 (* X1 X2))
+ 588.7139943790503 * (sin (cube (* X1 X3)))
+ 0.010578316285994994 * (cube (* X1 X3))
+ 10019.955215188047 * (- X3 (mydivide X4 X5))
+ -86.29951097746415 * (cos (square (- X3 X1)))
+ 0.035012855350998774 * (square (- X3 X1))
+ -3475.93354468703 * (cube (* X1 X2))
+ -494.65217888160436 * (* X1 X3)
+ 82.7112134609334 * (mydivide (mydivide X4 X3) X4)
+ -153693.3432149793 * (* (* X1 X3) X2)
+ 203.11933956602363 * (cos (cube (* X1 X3)))
+ 45875.3959883699 * (* X1 X2)
+ -3.881581577399716E-9 * (mydivide X5 (* X3 X2))

Progress Stalled, exiting. BEST MSE IS: 2059271.9574341103

Fastsr

Remark:
（数据格式不是 csv，在 excel 中复制到 txt 即可）
fastsr_paper.py

artifical_plus.txt

Score: 2542.081509036366
Best Individuals:
2542.08150904 : add(X0, add(add(cube(X3), add(multiply(X3, exp(numpy_protected_sqrt(X1))), cube(X3))), multiply(exp(X0), add(X2, X1))))
2964.89701394 : add(add(cube(X3), add(multiply(X0, X1), cube(X3))), multiply(exp(X0), add(X2, X1)))
3013.53941564 : add(add(square(add(X0, X0)), multiply(exp(X0), add(X2, X1))), cube(X3))
3200.86223946 : add(add(cube(X3), cube(X1)), multiply(exp(X0), add(X2, X1)))
3415.66476448 : add(cube(add(X0, X3)), X3)
3485.99329248 : cube(add(X0, X3))
6921.64162898 : square(square(X0))
9276.53681048 : cube(X0)
10210.7750205 : add(X0, add(X0, X0))
10619.6535005 : add(X0, X0)
10626.4650525 : numpy_protected_log_abs(square(exp(X2)))
11037.5319805 : X0

artifical_plus2.txt

Score: 129.134174737376
Best Individuals:
129.134174737 : subtract(subtract(subtract(X3, cube(X1)), X1), cube(cbrt(cube(X1))))
136.583417432 : subtract(cube(subtract(X3, X1)), cube(X1))
161.723155424 : subtract(subtract(X3, X1), cube(X1))
182.352272625 : subtract(numpy_protected_log_abs(X3), cube(X1))
186.320264738 : subtract(X3, cube(X1))
541.677526797 : subtract(subtract(X3, X1), X1)
579.787614466 : subtract(subtract(X2, X1), X0)
614.124636111 : subtract(X3, X1)
661.303220672 : numpy_protected_log_abs(X3)
679.28524297 : cbrt(X3)
682.420336276 : numpy_protected_sqrt(X3)
695.571745425 : X3

artifical_times.txt

Score: 1439.6021055178285
Best Individuals:
1439.60210552 : cube(numpy_protected_log_abs(square(add(X1, add(add(X3, X0), X0)))))
1999.56810032 : add(square(add(add(add(X3, X0), X1), X1)), X0)
2039.25770032 : square(add(X1, add(add(X3, X0), X0)))
3228.84234832 : cube(add(X3, X0))

6647.75803082 : square(square(X1))
7366.61986832 : add(add(add(X3, add(add(add(add(add(X3, X0), X0), X1), X0), X0)), X1), X0)
7712.45478032 : add(add(add(X3, add(add(add(add(X3, X0), X1), X0), X0)), X1), X0)
8067.28969232 : add(add(X3, add(add(add(add(X3, X0), X1), X0), X0)), X1)
8426.85593232 : add(add(add(X3, add(add(X3, X0), X0)), X1), X0)
8660.49075032 : cube(X0)
9178.06705232 : add(add(add(X3, X0), X1), X0)
9562.63329232 : add(add(X3, X0), X0)
9959.46820432 : add(X3, X0)
10112.4896284 : add(cbrt(X3), X0)
10363.8444123 : X0
10456.7492348 : cbrt(square(X1))
10521.3612654 : cbrt(X3)

Cd_wedge.txt
Score: 0.0024001156393774524
Best Individuals:
0.00240011563938 : subtract(multiply(X1, X0), X0)
0.00242580625339 : numpy_protected_sqrt(cube(X0))
0.00257609590654 : square(X0)
0.0032521474577 : subtract(X0, X0)
0.00834067856007 : X0
0.00834067856007 : numpy_protected_log_abs(exp(X0))

Heatflux_flatplate.txt
Score: 6.222447586898567e-06
Best Individuals:
6.2224475869e-06 : multiply(square(X0), add(X0, X0))
6.25340946164e-06 : cube(X0)
1.74682945075e-05 : subtract(X1, X1)
0.0109053443065 : X0
7818.94183676 : multiply(cbrt(X1), numpy_protected_sqrt(numpy_protected_log_abs(cube(X1))))

Heatflux_sphere.txt
Score: 32985930.491694402
Best Individuals:
32985930.4917 : exp(cbrt(subtract(X0, cbrt(exp(cbrt(subtract(X0, cbrt(X0))))))))
33617984.239 : exp(cbrt(subtract(X0, cbrt(cbrt(add(X3, square(X4)))))))
34133807.029 : exp(cbrt(subtract(X0, cbrt(X0))))
35295500.9437 : add(X1, exp(cbrt(subtract(X0, X2))))
35295501.0617 : exp(cbrt(subtract(X0, X2)))

35313089.2119 : exp(cbrt(X0))
47264747.5935 : subtract(add(add(subtract(X3, X3), add(X0, X2)), multiply(X0, cbrt(X0))), X2)
51307865.5365 : multiply(X0, cbrt(X0))
75054084.4149 : cbrt(add(X3, square(X4)))
120290548.838 : subtract(add(add(add(subtract(X3, X3), add(X0, X2)), add(X0, X2)), cbrt(X2)), X2)
133295911.868 : cbrt(square(X3))
134194787.177 : subtract(add(add(subtract(X3, X3), add(X0, X2)), cbrt(X2)), X2)
134200395.374 : add(X1, X0)
134200406.207 : X0
149271982.655 : square(numpy_protected_log_abs(add(cbrt(cbrt(add(X2, X1))), numpy_protected_sqrt(cbrt(numpy_protected_log_abs(cube(X4)))))))
149277164.892 : square(numpy_protected_log_abs(add(cbrt(cbrt(add(X2, X1))), numpy_protected_sqrt(cbrt(cbrt(X0))))))

FFX

FFX_paper.py

artifical_plus.txt

('Number of bases is: ', 0)
('nmse is: ', 24.98346751270535)
model is:
107
('Number of bases is: ', 1)
('nmse is: ', 11.983232756250928)
model is:
103 + 2.15*x1
('Number of bases is: ', 2)
('nmse is: ', 6.1084599453686685)
model is:
103 + 0.524*x1^2 + 0.356*x4 * x1
('Number of bases is: ', 3)
('nmse is: ', 4.488932443230234)
model is:
103 + 0.553*x1^2 + 0.393*x4 * x1 - 0.0434*x2^2
('Number of bases is: ', 4)
('nmse is: ', 3.807428650764011)
model is:
(100 - 0.651*log10(x3) - 0.293*x2) / (1.0 - 0.0274*x1 - 0.00834*x4)

('Number of bases is: ', 5)
('nmse is: ', 2.6883547606853484)
model is:
(99.5 - 3.41*log10(x1) - 0.721*log10(x3) - 0.307*x2) / (1.0 - 0.0346*x1 - 0.00836*x4)
('Number of bases is: ', 6)
('nmse is: ', 1.9366750961895705)
model is:
(99.5 - 5.45*log10(x1) - 0.744*log10(x3) - 0.717*x2) / (1.0 - 0.0387*x1 - 0.0162*log10(x2) - 0.00831*x4)
('Number of bases is: ', 7)
('nmse is: ', 1.0256312429579237)
model is:
98.5 - 10.4*log10(x1) + 5.65*x1 + 4.73*log10(x2) - 1.47*x2 + 0.994*x4 - 0.866*log10(x3) + 0.00195*log10(x4)
('Number of bases is: ', 9)
('nmse is: ', 1.0152266741361116)
model is:
102 + 0.926*x4 + 0.784*x1^2 + 0.454*x2 - 0.222*x2^2 - 0.198*x3 + 0.0177*x2 * x4 + 0.0146*x4 * x1 + 0.0143*x2 * x1 + 0.000326*x4^2
('Number of bases is: ', 13)
('nmse is: ', 0.909580081970452)
model is:
101 + 1.78*x1 + 1.13*max(0,x1-2.20) + 0.923*x4 + 0.736*max(0,x1-1.40) - 0.676*max(0,x2-2.20) - 0.539*log10(x3) + 0.385*max(0,x1-1.67) + 0.232*max(0,x1-2.47) - 0.140*max(0,x2-1.93) + 0.0481*max(0,2.20-x3) - 0.0281*max(0,2.73-x4) + 0.0102*max(0,1.67-x3) + 0.00308*max(0,x4-1.40)

artifical_plus2.txt

('Number of bases is: ', 0)
('nmse is: ', 26.60507268424809)
model is:
-12.1
('Number of bases is: ', 1)
('nmse is: ', 23.384884157799608)
model is:
-9.35 - 5.50*max(0,x2-1.67)
('Number of bases is: ', 2)
('nmse is: ', 6.846915943215394)
model is:
12.5 - 6.36*x2^2 + 0.997*x1 * x4
('Number of bases is: ', 3)
('nmse is: ', 4.115712744954907)
model is:
11.5 - 7.18*x2^2 + 1.67*x1 * x4 + 0.455*x4^2
('Number of bases is: ', 4)

('nmse is: ', 3.9940474808526374)
model is:
11.1 - 7.23*x2^2 + 1.70*x1 * x4 + 0.482*x4^2 + 0.159*x3
('Number of bases is: ', 5)
('nmse is: ', 3.3600290734278193)
model is:
8.27 - 7.44*x2^2 + 1.23*x4^2 + 0.952*x3 + 0.750*x1^2 + 0.565*x1 * x4
('Number of bases is: ', 6)
('nmse is: ', 2.361759219008762)
model is:
59.1 + 178*log10(x2) - 71.4*x2 + 6.32*x4 + 4.38*x1 + 2.58*log10(x3) + 0.617*x3
('Number of bases is: ', 7)
('nmse is: ', 2.2185101898227844)
model is:
59.6 + 186*log10(x2) - 73.3*x2 + 7.54*x4 - 5.11*log10(x4) + 4.40*x1 + 2.63*log10(x3) + 0.628*x3
('Number of bases is: ', 8)
('nmse is: ', 1.007967614542168)
model is:
-13.5 + 22.8*x2 - 13.1*x2^2 + 1.56*x4^2 + 1.28*x3 + 1.08*x1^2 + 0.0440*x4 * x2 + 0.0390*x1 * x2 + 0.0270*x1 * x4
('Number of bases is: ', 16)
('nmse is: ', 0.9725610795220343)
model is:
4.84 - 15.4*max(0,x2-2.20) - 10.8*max(0,x2-2.47) - 9.05*x2 - 7.47*max(0,x2-1.40) - 6.28*max(0,x2-1.67) + 3.32*x4 - 2.26*max(0,x2-1.93) + 2.25*x1 + 1.99*max(0,x4-2.20) + 1.37*max(0,x4-1.67) + 1.30*max(0,x1-1.67) + 1.18*max(0,x4-1.40) + 1.03*max(0,x1-2.20) + 0.651*max(0,x1-1.40) - 0.578*max(0,2.73-x3) + 0.425*x3
('Number of bases is: ', 17)
('nmse is: ', 0.9712551944214991)
model is:
4.07 - 15.4*max(0,x2-2.20) - 10.8*max(0,x2-2.47) - 9.05*x2 - 7.47*max(0,x2-1.40) - 6.28*max(0,x2-1.67) + 3.32*x4 - 2.26*max(0,x2-1.93) + 2.25*x1 + 1.99*max(0,x4-2.20) + 1.69*log10(x3) + 1.37*max(0,x4-1.67) + 1.30*max(0,x1-1.67) + 1.18*max(0,x4-1.40) + 1.03*max(0,x1-2.20) + 0.651*max(0,x1-1.40) + 0.415*x3 - 0.145*max(0,2.73-x3)

artifical_times.txt

('Number of bases is: ', 0)
('nmse is: ', 17.137290900859234)
model is:
104
('Number of bases is: ', 1)
('nmse is: ', 14.079753525490277)
model is:
102 + 0.482*x4 * x1

('Number of bases is: ', 2)
('nmse is: ', 13.55087856203186)
model is:
101 + 0.545*x4 * x1 + 0.0403*x1^2
('Number of bases is: ', 3)
('nmse is: ', 9.978370362990091)
model is:
101 + 0.774*x4 * x1 - 0.260*x2^2 + 0.194*x1^2
('Number of bases is: ', 4)
('nmse is: ', 8.024439755999206)
model is:
102 + 0.946*x4 * x1 - 0.784*x3 - 0.458*x2^2 + 0.309*x1^2
('Number of bases is: ', 5)
('nmse is: ', 7.595292847702499)
model is:
102 - 1.45*max(0,x2-2.20) * x1 + 1.34*max(0,x1-1.40) * x4 - 0.997*log10(x3) - 0.326*max(0,x2-2.20) * x4 + 0.321*x4 * x1
('Number of bases is: ', 6)
('nmse is: ', 7.125656726523736)
model is:
102 - 1.41*log10(x3) + 1.38*max(0,x1-1.40) * x4 - 1.35*max(0,x2-2.20) * x1 - 0.554*max(0,x1-1.40) * max(0,x2-2.47) - 0.428*max(0,x2-2.20) * x4 + 0.343*x4 * x1
('Number of bases is: ', 7)
('nmse is: ', 5.192526716187131)
model is:
93.6 + 9.46*x2 - 1.76*x2^2 + 1.65*x4 * x1 - 1.48*x2 * x1 - 1.01*x3 + 0.743*x1^2 - 0.720*x4 * x2
('Number of bases is: ', 8)
('nmse is: ', 5.141278010502279)
model is:
93.4 + 9.85*x2 - 1.83*x2^2 + 1.67*x4 * x1 - 1.51*x2 * x1 - 1.21*x3 + 0.753*x1^2 - 0.739*x4 * x2 + 0.0477*x3^2
('Number of bases is: ', 9)
('nmse is: ', 4.99729593774279)
model is:
92.4 + 11.5*x2 - 2.22*x3 - 2.09*x2^2 + 1.67*x4 * x1 - 1.65*x2 * x1 - 0.892*x4 * x2 + 0.820*x1^2 + 0.308*x4 + 0.297*x3^2
('Number of bases is: ', 11)
('nmse is: ', 4.975495653684135)
model is:
91.8 + 12.3*x2 - 2.67*x3 - 2.22*x2^2 - 1.71*x2 * x1 + 1.67*x4 * x1 - 0.962*x4 * x2 + 0.833*x1^2 + 0.461*x4 + 0.409*x3^2 + 0.0933*x1 + 0.000269*x4^2
('Number of bases is: ', 12)
('nmse is: ', 4.448755888716317)
model is:

(93.4 - 7.60*x4 - 6.08*x1 - 0.822*x2^2 - 0.815*x1 - 0.748*x1 * x2 - 0.405*x1 * x3 - 0.178*x4^2) / (1.0 - 0.0918*x1 - 0.0893*x4 - 0.0345*x2 - 0.0135*x1 - 0.00161*x3 * x2)
('Number of bases is: ', 13)
('nmse is: ', 3.900818102347504)
model is:
(94.3 - 9.06*x4 - 7.83*x1 - 0.794*x2^2 - 0.659*x1 * x2 - 0.633*x1 - 0.341*x1 * x3 - 0.200*x4^2 - 0.0992*x3) / (1.0 - 0.107*x1 - 0.103*x4 - 0.0325*x2 - 0.00935*x1 - 0.00171*x3 * x2)
('Number of bases is: ', 16)
('nmse is: ', 3.6881671272707264)
model is:
(94.6 - 9.27*x4 - 8.23*x1 - 1.06*x2 - 0.760*x2^2 - 0.641*x1 * x2 - 0.565*x1 - 0.310*x1 * x3 - 0.203*x4^2 - 0.144*x3 - 0.0187*x1^2 - 0.00120*x1^2) / (1.0 - 0.110*x1 - 0.105*x4 - 0.0416*x2 - 0.00818*x1 - 0.00176*x3 * x2)
('Number of bases is: ', 17)
('nmse is: ', 3.6666916491947656)
model is:
(95.2 - 9.48*x4 - 8.63*x1 - 2.31*x2 - 0.682*x2^2 - 0.610*x1 * x2 - 0.500*x1 - 0.283*x1 * x3 - 0.211*x3 - 0.206*x4^2 - 0.0478*x1^2 - 0.00170*x1^2) / (1.0 - 0.113*x1 - 0.107*x4 - 0.0505*x2 - 0.00710*x1 - 0.00178*x3 * x2 - 8.01e-5*x3^2)
('Number of bases is: ', 20)
('nmse is: ', 3.6281822258348986)
model is:
100 + 4.06*log10(x3) * max(0,x2-2.20) - 3.83*max(0,x1-1.40) * max(0,x2-2.20) - 1.83*log10(x3) * x1 - 1.51*max(0,x1-2.47) * log10(x3) - 1.02*max(0,x2-2.20) * x4 - 0.962*max(0,x1-2.47) * max(0,x2-2.20) + 0.929*x4 * x1 + 0.773*max(0,x1-1.40) * x4 - 0.707*log10(x3) * x4 - 0.657*max(0,1.40-x2) - 0.547*max(0,x1-1.40) * max(0,x2-1.93) + 0.527*max(0,x1-2.47) * x4 - 0.371*max(0,1.40-x2) * x1 + 0.371*max(0,x1-1.93) - 0.272*max(0,x2-1.67) * x4 + 0.203*x1^2 - 0.181*max(0,x2-1.93) * x4 + 0.152*max(0,x1-1.40) * x1 - 0.0721*max(0,x1-2.47) * max(0,x2-2.47) + 0.0438*max(0,x1-2.20)
('Number of bases is: ', 21)
('nmse is: ', 3.5356567893051567)
model is:
100 + 4.31*log10(x3) * max(0,x2-2.20) - 3.67*max(0,x1-1.40) * max(0,x2-2.20) - 1.82*log10(x3) * x1 - 1.81*max(0,x1-2.47) * log10(x3) - 0.993*max(0,x2-2.20) * x4 + 0.944*x4 * x1 - 0.901*max(0,x1-2.47) * max(0,x2-2.20) + 0.749*max(0,x1-1.40) * x4 - 0.737*log10(x3) * x4 - 0.698*max(0,x1-1.40) * max(0,x2-1.93) - 0.580*max(0,1.40-x2) + 0.549*max(0,x1-2.47) * x4 - 0.435*max(0,1.40-x2) * x1 + 0.330*max(0,x1-1.93) - 0.279*max(0,x2-1.67) * x4 - 0.224*max(0,x1-2.47) * max(0,x2-2.47) + 0.201*x1^2 - 0.195*max(0,x2-1.93) * x4 + 0.171*max(0,x1-1.40) * x1 + 0.0997*max(0,x1-2.20) - 0.0443*max(0,x2-2.47) * x4
('Number of bases is: ', 23)
('nmse is: ', 3.167193307240467)
model is:
100 + 5.31*log10(x3) * max(0,x2-2.20) - 2.99*max(0,x1-2.47) * log10(x3) - 2.65*max(0,x1-1.40) * max(0,x2-2.20) - 1.78*log10(x3) * x1 - 1.45*max(0,x1-1.40) * max(0,x2-1.93) + 1.01*x4 * x1 - 0.862*log10(x3) * x4 - 0.843*max(0,x2-2.20) * x4 - 0.838*max(0,x1-2.47) * max(0,x2-2.20) - 0.689*max(0,1.40-x2) * x1 + 0.656*max(0,x1-2.47) * x4 + 0.638*max(0,x1-1.40) * x4 - 0.548*max(0,x1-2.47) * max(0,x2-2.47) +

0.545*max(0,x1-2.20) + 0.454*max(0,x1-1.67) - 0.299*max(0,x2-1.67) * x4 - 0.281*max(0,x2-1.93) * x4 - 0.242*max(0,x1-1.40) * max(0,x2-2.47) - 0.227*max(0,x2-2.47) * x4 + 0.199*x1^2 - 0.183*max(0,1.40-x2) + 0.111*max(0,x1-1.40) * x1 - 0.0448*max(0,1.40-x2) * x4
('Number of bases is: ', 25)
('nmse is: ', 3.013755006166342)
model is:
100 + 5.71*log10(x3) * max(0,x2-2.20) - 3.47*max(0,x1-2.47) * log10(x3) - 2.10*max(0,x1-1.40) * max(0,x2-2.20) - 1.80*max(0,x1-1.40) * max(0,x2-1.93) - 1.76*log10(x3) * x1 + 1.05*x4 * x1 - 0.925*max(0,x1-2.47) * max(0,x2-2.20) - 0.917*log10(x3) * x4 - 0.773*max(0,x2-2.20) * x4 + 0.715*max(0,x1-2.20) - 0.714*max(0,1.40-x2) * x1 + 0.693*max(0,x1-2.47) * x4 + 0.591*max(0,x1-1.40) * x4 - 0.525*max(0,x1-2.47) * max(0,x2-2.47) + 0.459*max(0,x1-1.67) - 0.457*max(0,x1-1.40) * max(0,x2-2.47) - 0.329*max(0,x2-1.93) * x4 - 0.307*max(0,x2-1.67) * x4 - 0.294*max(0,x2-2.47) * x4 + 0.176*x1^2 + 0.139*max(0,x1-1.40) - 0.105*max(0,1.40-x2) * x4 + 0.0973*max(0,x1-1.40) * x1 - 0.0865*max(0,x1-1.40) * max(0,1.40-x2) - 0.0401*max(0,1.40-x2)
('Number of bases is: ', 26)
('nmse is: ', 2.934964592035354)
model is:
100 + 6.05*log10(x3) * max(0,x2-2.20) - 3.79*max(0,x1-2.47) * log10(x3) - 2.14*max(0,x1-1.40) * max(0,x2-1.93) - 1.78*log10(x3) * x1 - 1.55*max(0,x1-1.40) * max(0,x2-2.20) + 1.08*x4 * x1 - 1.02*max(0,x1-2.47) * max(0,x2-2.20) - 0.978*log10(x3) * x4 + 0.849*max(0,x1-2.20) + 0.739*max(0,x1-2.47) * x4 - 0.705*max(0,x2-2.20) * x4 - 0.674*max(0,x1-1.40) * max(0,x2-2.47) + 0.541*max(0,x1-1.40) * x4 - 0.483*max(0,x1-2.47) * max(0,x2-2.47) - 0.481*max(0,1.40-x2) * x1 - 0.473*max(0,x1-1.40) * max(0,1.40-x2) + 0.422*max(0,x1-1.67) + 0.381*max(0,x1-1.40) - 0.378*max(0,x2-1.93) * x4 - 0.354*max(0,x2-2.47) * x4 - 0.309*max(0,x2-1.67) * x4 - 0.217*max(0,1.40-x2) - 0.154*max(0,1.40-x2) * x4 + 0.144*x1^2 + 0.0808*max(0,x1-1.40) * x1 + 0.00585*x3^2
('Number of bases is: ', 29)
('nmse is: ', 2.7764546828341663)
model is:
100 + 5.83*log10(x3) * max(0,x2-2.20) - 3.89*max(0,x1-2.47) * log10(x3) - 2.47*max(0,x1-1.40) * max(0,x2-1.93) - 1.88*log10(x3) * x1 + 1.12*x4 * x1 - 1.07*log10(x3) * x4 - 1.06*max(0,x1-2.47) * max(0,x2-2.20) - 1.04*max(0,x1-1.40) * max(0,x2-2.20) + 0.890*max(0,x1-2.20) - 0.883*max(0,x1-1.40) * max(0,x2-2.47) - 0.824*max(0,x1-1.40) * max(0,1.40-x2) + 0.777*max(0,x1-2.47) * x4 - 0.654*max(0,x2-2.20) * x4 + 0.637*max(0,x1-1.40) + 0.474*max(0,x1-1.40) * x4 - 0.459*max(0,x2-1.93) * x4 - 0.448*max(0,x1-2.47) * max(0,x2-2.47) + 0.425*max(0,x1-1.67) - 0.410*max(0,x2-2.47) * x4 - 0.306*max(0,x2-1.67) * x4 - 0.284*max(0,1.40-x2) * x1 + 0.276*log10(x3) * max(0,x2-1.93) - 0.259*max(0,1.40-x2) - 0.237*max(0,1.40-x2) * x4 + 0.183*max(0,x2-2.20) + 0.110*x1^2 + 0.0597*max(0,x1-1.40) * x1 + 0.0418*max(0,x1-1.93) + 0.0273*x3^2
('Number of bases is: ', 32)
('nmse is: ', 2.6290040135495074)
model is:
100 + 5.39*log10(x3) * max(0,x2-2.20) - 4.00*max(0,x1-2.47) * log10(x3) - 2.78*max(0,x1-1.40) * max(0,x2-1.93) - 1.96*log10(x3) * x1 + 1.17*x4 * x1 - 1.15*log10(x3) * x4 - 1.14*max(0,x1-1.40) * max(0,1.40-x2) - 1.08*max(0,x1-2.47) * max(0,x2-2.20) - 1.08*max(0,x1-1.40) * max(0,x2-2.47) + 0.874*max(0,x1-1.40) + 0.800*max(0,x1-2.47) * x4 + 0.701*max(0,x1-2.20) + 0.642*log10(x3) * max(0,x2-1.93) - 0.608*max(0,x2-

2.20) * x4 - 0.595*max(0,x1-1.40) * max(0,x2-2.20) - 0.557*max(0,x2-1.93) * x4 - 0.465*max(0,x2-2.47) * x4 + 0.453*max(0,x2-2.20) + 0.442*max(0,x1-1.67) + 0.412*max(0,x1-1.40) * x4 - 0.389*max(0,x1-2.47) * max(0,x2-2.47) - 0.332*max(0,1.40-x2) * x4 - 0.300*max(0,x2-1.67) * x4 - 0.231*max(0,1.40-x2) + 0.226*max(0,x1-1.93) - 0.114*max(0,1.40-x2) * x1 + 0.105*max(0,x1-2.73) + 0.0728*x1^2 + 0.0455*x3^2 + 0.0425*max(0,x1-2.47) + 0.0365*max(0,x1-1.40) * x1 + 0.0109*max(0,x1-2.47) * x1
('Number of bases is: ', 33)
('nmse is: ', 2.4222562795347424)
model is:
98.8 + 3.88*log10(x3) * max(0,x2-2.20) - 3.62*max(0,x1-2.47) * log10(x3) - 3.24*max(0,x1-1.40) * max(0,x2-1.93) - 2.51*log10(x3) * x1 + 1.90*log10(x3) * max(0,x2-1.93) - 1.57*log10(x3) * x4 - 1.53*max(0,x1-1.40) * max(0,1.40-x2) - 1.39*max(0,x1-1.40) * max(0,x2-2.47) + 1.17*max(0,x2-2.20) + 1.14*x4 * x1 + 1.07*max(0,x1-1.40) - 0.842*max(0,x2-1.93) * x4 + 0.770*max(0,x1-2.47) * x4 + 0.763*log10(x3) * max(0,1.40-x2) - 0.713*max(0,x1-2.47) * max(0,x2-2.47) + 0.632*max(0,x1-1.93) - 0.588*max(0,x2-2.47) * x4 + 0.574*max(0,x1-2.47) - 0.571*max(0,1.40-x2) * x4 - 0.563*max(0,x1-2.47) * max(0,x2-1.67) + 0.499*x3 - 0.479*max(0,x2-2.20) * x4 + 0.454*max(0,x1-1.40) * x4 + 0.377*max(0,x1-1.67) + 0.313*x1 - 0.284*max(0,x2-1.67) * x4 - 0.276*max(0,x1-2.47) * max(0,1.40-x2) + 0.257*x4 + 0.213*log10(x3) + 0.184*max(0,x1-2.20) - 0.0547*max(0,2.47-x3) + 0.0179*x1^2 + 0.00296*x3^2
('Number of bases is: ', 34)
('nmse is: ', 2.393128943967868)
model is:
98.6 - 3.46*max(0,x1-2.47) * log10(x3) - 3.25*max(0,x1-1.40) * max(0,x2-1.93) + 3.10*log10(x3) * max(0,x2-2.20) - 2.64*log10(x3) * x1 + 2.40*log10(x3) * max(0,x2-1.93) - 1.67*log10(x3) * x4 - 1.54*max(0,x1-1.40) * max(0,1.40-x2) - 1.42*max(0,x1-1.40) * max(0,x2-2.47) + 1.29*max(0,x2-2.20) + 1.09*x4 * x1 + 1.05*log10(x3) * max(0,1.40-x2) + 0.988*max(0,x1-1.40) - 0.870*max(0,x2-1.93) * x4 + 0.762*max(0,x1-2.47) * x4 + 0.681*max(0,x1-1.93) - 0.635*max(0,x1-2.47) * max(0,x2-1.67) + 0.627*log10(x3) + 0.614*max(0,x1-2.47) - 0.611*max(0,1.40-x2) * x4 - 0.611*max(0,x2-2.47) * x4 - 0.569*max(0,x1-2.47) * max(0,x2-2.47) + 0.508*max(0,x1-1.40) * x4 + 0.477*x3 - 0.454*max(0,x2-2.20) * x4 + 0.449*x1 - 0.405*max(0,x1-2.47) * max(0,1.40-x2) + 0.361*x4 + 0.347*max(0,x1-1.67) - 0.298*max(0,x2-1.67) * x4 + 0.245*log10(x3) * max(0,x2-2.47) + 0.127*max(0,x1-2.20) - 0.0607*max(0,2.47-x3) + 0.0137*x1^2 + 0.00634*x3^2
('Number of bases is: ', 36)
('nmse is: ', 2.378149796555744)
model is:
98.4 - 3.33*max(0,x1-2.47) * log10(x3) - 3.26*max(0,x1-1.40) * max(0,x2-1.93) + 2.83*log10(x3) * max(0,x2-1.93) - 2.75*log10(x3) * x1 + 2.35*log10(x3) * max(0,x2-2.20) - 1.75*log10(x3) * x4 - 1.54*max(0,x1-1.40) * max(0,1.40-x2) - 1.45*max(0,x1-1.40) * max(0,x2-2.47) + 1.29*log10(x3) * max(0,1.40-x2) + 1.16*max(0,x2-2.20) + 1.05*x4 * x1 + 0.988*log10(x3) + 0.924*max(0,x1-1.40) - 0.880*max(0,x2-1.93) * x4 + 0.754*max(0,x1-2.47) * x4 + 0.735*max(0,x1-1.93) - 0.696*max(0,x1-2.47) * max(0,x2-1.67) + 0.662*max(0,x1-2.47) - 0.646*max(0,1.40-x2) * x4 - 0.625*max(0,x2-2.47) * x4 + 0.583*x1 + 0.580*log10(x3) * max(0,x2-2.47) + 0.556*max(0,x1-1.40) * x4 - 0.523*max(0,x1-2.47) * max(0,1.40-x2) + 0.455*x3 + 0.454*x4 - 0.451*max(0,x1-2.47) * max(0,x2-2.47) - 0.429*max(0,x2-2.20) * x4 - 0.328*max(0,x2-1.67) * x4 + 0.320*max(0,x1-1.67) + 0.144*max(0,x2-1.93) + 0.0713*max(0,x1-2.20) - 0.0661*max(0,2.47-x3) + 0.0540*max(0,x2-2.47) + 0.00968*x3^2 + 0.00382*x1^2
('Number of bases is: ', 40)
('nmse is: ', 2.347451806561142)

model is:
98.1 - 3.22*max(0,x1-2.47) * log10(x3) - 3.07*max(0,x1-1.40) * max(0,x2-1.93) - 2.85*log10(x3) * x1 + 2.60*log10(x3) * max(0,x2-1.93) + 2.00*log10(x3) * max(0,x2-2.20) - 1.83*log10(x3) * x4 - 1.62*max(0,x1-1.40) * max(0,1.40-x2) + 1.56*log10(x3) * max(0,1.40-x2) - 1.45*max(0,x1-1.40) * max(0,x2-2.47) + 1.25*log10(x3) + 1.01*x4 * x1 - 1.01*max(0,x2-1.93) * x4 + 0.863*max(0,x2-2.20) + 0.863*log10(x3) * max(0,x2-2.47) + 0.817*max(0,x1-1.40) + 0.766*max(0,x1-1.93) + 0.757*x1 + 0.746*max(0,x1-2.47) * x4 - 0.692*max(0,x1-2.47) * max(0,x2-1.67) + 0.676*max(0,x1-2.47) - 0.664*max(0,1.40-x2) * x4 - 0.608*max(0,x2-2.47) * x4 + 0.602*max(0,x2-1.93) + 0.600*max(0,x1-1.40) * x4 - 0.566*max(0,x1-2.47) * max(0,x2-2.47) + 0.529*x4 - 0.517*max(0,x1-2.47) * max(0,1.40-x2) + 0.436*x3 - 0.398*max(0,x2-2.20) * x4 + 0.328*max(0,x2-1.67) * log10(x3) - 0.272*max(0,x2-1.67) * x4 + 0.271*max(0,x1-1.67) - 0.143*max(0,x2-1.67) * x1 - 0.0671*max(0,2.47-x3) + 0.0596*max(0,x2-2.47) + 0.0502*max(0,x1-2.20) + 0.0248*max(0,x2-1.40) * max(0,x2-2.20) + 0.0131*x3^2 - 0.00760*max(0,2.20-x3) + 0.00159*x1^2
('Number of bases is: ', 43)
('nmse is: ', 2.326338925094848)
model is:
97.9 - 3.12*max(0,x1-2.47) * log10(x3) - 2.93*log10(x3) * x1 - 2.89*max(0,x1-1.40) * max(0,x2-1.93) + 2.28*log10(x3) * max(0,x2-1.93) - 1.89*log10(x3) * x4 + 1.82*log10(x3) * max(0,1.40-x2) - 1.70*max(0,x1-1.40) * max(0,1.40-x2) + 1.67*log10(x3) * max(0,x2-2.20) - 1.49*max(0,x1-1.40) * max(0,x2-2.47) + 1.44*log10(x3) - 1.12*max(0,x2-1.93) * x4 + 1.10*log10(x3) * max(0,x2-2.47) + 0.980*x4 * x1 + 0.905*x1 + 0.836*max(0,x2-1.93) + 0.797*max(0,x1-1.93) + 0.740*max(0,x1-2.47) * x4 + 0.736*max(0,x2-1.67) * log10(x3) + 0.720*max(0,x1-1.40) - 0.697*max(0,x1-2.47) * max(0,x2-1.67) + 0.687*max(0,x1-2.47) - 0.682*max(0,1.40-x2) * x4 + 0.645*max(0,x2-2.20) + 0.638*max(0,x1-1.40) * x4 - 0.629*max(0,x1-2.47) * max(0,x2-2.47) - 0.624*max(0,x2-2.47) * x4 + 0.592*x4 - 0.507*max(0,x1-2.47) * max(0,1.40-x2) + 0.419*x3 - 0.353*max(0,x2-2.20) * x4 - 0.272*max(0,x2-1.67) * x1 + 0.229*max(0,x1-1.67) - 0.217*max(0,x2-1.67) * x4 + 0.0815*max(0,x2-1.40) * max(0,x2-1.93) - 0.0633*max(0,2.47-x3) + 0.0426*max(0,x2-1.67) * max(0,x2-1.93) + 0.0350*max(0,x2-1.40) * max(0,x2-2.20) + 0.0251*max(0,x1-2.20) - 0.0235*max(0,2.20-x3) + 0.0169*x3^2 + 0.00106*x1^2 - 0.000350*max(0,2.73-x4) + 3.63e-5*x4^2
('Number of bases is: ', 45)
('nmse is: ', 2.3113277661134872)
model is:
97.7 - 3.02*max(0,x1-2.47) * log10(x3) - 3.01*log10(x3) * x1 - 2.75*max(0,x1-1.40) * max(0,x2-1.93) + 2.31*log10(x3) * max(0,x2-1.93) + 2.01*log10(x3) * max(0,1.40-x2) - 1.94*log10(x3) * x4 - 1.75*max(0,x1-1.40) * max(0,1.40-x2) + 1.64*log10(x3) - 1.39*max(0,x1-1.40) * max(0,x2-2.47) + 1.33*log10(x3) * max(0,x2-2.47) - 1.27*max(0,x2-1.93) * x4 + 1.22*log10(x3) * max(0,x2-2.20) + 1.05*x1 + 0.951*x4 * x1 + 0.927*max(0,x2-1.93) + 0.915*max(0,x2-1.67) * log10(x3) + 0.823*max(0,x1-1.93) + 0.738*max(0,x1-2.47) * x4 - 0.719*max(0,x2-2.47) * x4 - 0.711*max(0,x1-2.47) * max(0,x2-1.67) - 0.707*max(0,x1-2.47) * max(0,x2-2.47) + 0.697*max(0,x1-2.47) - 0.696*max(0,1.40-x2) * x4 + 0.670*max(0,x1-1.40) * x4 + 0.644*x4 + 0.625*max(0,x1-1.40) - 0.517*max(0,x1-2.47) * max(0,1.40-x2) + 0.494*max(0,x2-2.20) + 0.396*x3 - 0.296*max(0,x2-1.67) * x1 - 0.210*max(0,x2-2.20) * x4 + 0.205*max(0,x2-1.67) * max(0,x2-1.93) + 0.194*max(0,x1-1.67) - 0.162*max(0,x2-1.67) * x4 + 0.127*max(0,x2-1.40) * max(0,x2-1.93) - 0.122*max(0,x2-2.20) * x1 - 0.0605*max(0,2.47-x3) - 0.0378*max(0,2.20-x3) - 0.0314*max(0,x2-1.40) * x1 + 0.0313*max(0,x2-1.40) * max(0,x2-2.20) + 0.0212*x3^2 - 0.0182*max(0,x2-2.47) * x1 + 0.00346*max(0,x1-2.20) - 0.00237*max(0,2.73-x4) + 0.000414*x4^2
('Number of bases is: ', 46)

('nmse is: ', 2.3000588700464357)
model is:
97.5 - 3.07*log10(x3) * x1 - 2.95*max(0,x1-2.47) * log10(x3) - 2.62*max(0,x1-1.40) * max(0,x2-1.93) + 2.50*log10(x3) * max(0,x2-1.93) + 2.13*log10(x3) * max(0,1.40-x2) - 1.98*log10(x3) * x4 + 1.81*log10(x3) - 1.78*max(0,x1-1.40) * max(0,1.40-x2) + 1.59*log10(x3) * max(0,x2-2.47) - 1.37*max(0,x2-1.93) * x4 + 1.19*x1 - 1.08*max(0,x1-1.40) * max(0,x2-2.47) + 0.963*max(0,x2-1.67) * log10(x3) + 0.959*max(0,x2-1.93) + 0.923*x4 * x1 + 0.839*max(0,x1-1.93) - 0.777*max(0,x1-2.47) * max(0,x2-1.67) - 0.750*max(0,x1-2.47) * max(0,x2-2.47) + 0.745*max(0,x1-2.47) + 0.740*log10(x3) * max(0,x2-2.20) + 0.728*max(0,x1-2.47) * x4 - 0.720*max(0,1.40-x2) * x4 - 0.714*max(0,x2-2.47) * x4 + 0.704*max(0,x1-1.40) * x4 + 0.698*x4 - 0.580*max(0,x1-2.47) * max(0,1.40-x2) + 0.508*max(0,x1-1.40) + 0.377*x3 + 0.351*max(0,x2-2.20) + 0.340*max(0,x2-1.67) * max(0,x2-1.93) - 0.281*max(0,x2-2.47) * x1 + 0.265*max(0,x2-1.40) * max(0,x2-1.93) - 0.208*max(0,x2-1.67) * x1 + 0.156*max(0,x1-1.67) - 0.143*max(0,x2-1.67) * x4 - 0.135*max(0,x2-2.20) * x1 - 0.133*max(0,x2-2.20) * x4 - 0.110*max(0,x2-1.93) * x1 - 0.101*max(0,x2-1.40) * x1 - 0.0581*max(0,2.47-x3) - 0.0500*max(0,2.20-x3) + 0.0276*max(0,x2-1.40) * max(0,x2-2.20) + 0.0249*x3^2 - 0.00378*max(0,2.73-x4) + 0.00333*max(0,x2-2.47) + 0.000681*x4^2
('Number of bases is: ', 47)
('nmse is: ', 2.2825327789615444)
model is:
97.2 - 3.16*log10(x3) * x1 + 3.06*log10(x3) * max(0,x2-1.93) - 2.84*max(0,x1-2.47) * log10(x3) + 2.37*log10(x3) * max(0,1.40-x2) - 2.27*max(0,x1-1.40) * max(0,x2-1.93) + 2.08*log10(x3) - 2.05*log10(x3) * x4 + 1.84*log10(x3) * max(0,x2-2.47) - 1.81*max(0,x1-1.40) * max(0,1.40-x2) - 1.45*max(0,x2-1.93) * x4 + 1.44*x1 + 0.952*max(0,x2-1.93) - 0.915*max(0,x1-2.47) * max(0,x2-1.67) + 0.877*x4 * x1 + 0.853*max(0,x1-1.93) + 0.829*max(0,x1-2.47) - 0.796*max(0,x1-2.47) * max(0,x2-2.47) + 0.785*x4 + 0.759*max(0,x1-1.40) * x4 - 0.754*max(0,1.40-x2) * x4 - 0.747*max(0,x2-2.47) * x4 + 0.714*max(0,x1-2.47) * x4 - 0.693*max(0,x1-2.47) * max(0,1.40-x2) - 0.669*max(0,x2-2.47) * x1 - 0.651*max(0,x1-1.40) * max(0,x2-2.47) + 0.636*max(0,x2-1.67) * log10(x3) + 0.630*max(0,x2-1.67) * max(0,x2-1.93) + 0.510*max(0,x2-1.40) * max(0,x2-1.93) - 0.481*max(0,x2-1.93) * x1 + 0.344*x3 + 0.288*max(0,x1-1.40) + 0.190*max(0,x2-1.40) * log10(x3) - 0.176*max(0,x2-1.40) * x1 + 0.163*max(0,x2-2.20) - 0.134*max(0,x2-1.67) * x4 + 0.0894*max(0,x1-1.67) - 0.0791*max(0,x2-2.20) * x1 - 0.0699*max(0,x2-1.67) * x1 - 0.0690*max(0,2.20-x3) - 0.0679*max(0,x2-2.20) * x4 - 0.0546*max(0,2.47-x3) + 0.0539*log10(x3) * max(0,x2-2.20) + 0.0310*x3^2 + 0.0126*max(0,x1-2.20) + 0.0108*max(0,x2-1.40) * max(0,x2-2.20) - 0.00528*max(0,2.73-x4) + 0.000975*x4^2
('Number of bases is: ', 50)
('nmse is: ', 2.277470123502084)
model is:
96.9 + 4.28*log10(x3) * max(0,x2-1.93) - 3.25*log10(x3) * x1 - 2.73*max(0,x1-2.47) * log10(x3) + 2.41*log10(x3) + 2.25*log10(x3) * max(0,1.40-x2) - 2.12*log10(x3) * x4 - 1.78*max(0,x1-1.40) * max(0,x2-1.93) - 1.71*max(0,x1-1.40) * max(0,1.40-x2) + 1.59*x1 - 1.47*max(0,x2-1.93) * x4 - 1.36*max(0,x2-2.73) + 1.35*max(0,x2-2.20) * max(0,x2-1.93) + 1.18*log10(x3) * max(0,x2-2.47) - 1.06*max(0,x2-1.93) * x1 - 1.04*max(0,x1-2.47) * max(0,x2-1.67) - 0.995*max(0,x2-2.47) * x1 - 0.970*max(0,x1-2.47) * max(0,1.40-x2) - 0.913*max(0,x2-2.47) * x4 + 0.867*max(0,x1-2.47) + 0.852*x4 + 0.842*x4 * x1 + 0.815*max(0,x1-1.93) + 0.798*max(0,x1-1.40) * x4 - 0.781*max(0,1.40-x2) * x4 + 0.716*max(0,x1-2.47) * x4 + 0.679*max(0,x2-1.67) * max(0,x2-1.93) + 0.517*max(0,x2-1.40) * max(0,x2-1.93) + 0.496*max(0,x2-1.93) - 0.462*max(0,x1-2.47) * max(0,x2-2.47) - 0.453*max(0,x1-1.40) * max(0,x2-2.47) + 0.339*x3 + 0.230*max(0,x2-1.40) * max(0,x2-1.67)

- 0.156*max(0,x2-1.40) * max(0,x1-2.47) - 0.112*max(0,x2-1.67) * x4 + 0.0962*max(0,x1-2.20) - 0.0877*max(0,x2-1.40) * x1 - 0.0821*max(0,2.20-x3) + 0.0782*max(0,x1-1.40) + 0.0702*max(0,x2-1.40) * log10(x3) + 0.0664*max(0,1.93-x2) - 0.0504*max(0,2.47-x3) + 0.0401*log10(x3) * max(0,x2-2.20) + 0.0396*max(0,x1-2.73) + 0.0329*x3^2 + 0.0220*max(0,x2-1.67) * log10(x3) + 0.0167*x1^2 - 0.00752*max(0,x2-1.40) * x4 - 0.00584*max(0,2.73-x4) + 0.00431*max(0,x1-1.67) + 0.00111*x4^2
('Number of bases is: ', 52)
('nmse is: ', 2.2737208867833334)
model is:
96.8 + 3.49*log10(x3) * max(0,x2-1.93) - 3.30*log10(x3) * x1 - 2.67*max(0,x1-2.47) * log10(x3) - 2.63*max(0,x2-2.47) + 2.56*log10(x3) * max(0,1.40-x2) + 2.53*log10(x3) - 2.45*max(0,x2-2.73) + 2.29*max(0,x2-2.20) * max(0,x2-1.93) - 2.16*log10(x3) * x4 - 1.85*max(0,x1-1.40) * max(0,x2-1.93) - 1.76*max(0,x1-1.40) * max(0,1.40-x2) + 1.61*x1 + 1.58*log10(x3) * max(0,x2-2.47) + 1.47*max(0,x2-2.47) * max(0,x2-1.93) - 1.26*max(0,x2-1.93) * x4 - 1.03*max(0,x2-1.93) * x1 - 0.999*max(0,x1-2.47) * max(0,1.40-x2) - 0.919*max(0,x2-2.47) * x4 + 0.916*x4 + 0.884*max(0,x1-2.47) - 0.873*max(0,1.40-x2) * x4 + 0.825*x4 * x1 + 0.818*max(0,x1-1.40) * x4 + 0.810*max(0,x1-1.93) + 0.804*max(0,x2-1.67) * max(0,x2-1.93) - 0.769*max(0,x1-1.40) * max(0,x2-2.47) - 0.763*max(0,x1-2.47) * max(0,x2-1.67) + 0.712*max(0,x1-2.47) * x4 - 0.686*max(0,x2-2.47) * x1 - 0.487*max(0,x1-2.47) * max(0,x2-2.47) + 0.398*max(0,x2-1.40) * max(0,x2-1.93) + 0.377*max(0,x2-1.40) * log10(x3) - 0.345*max(0,x2-1.40) * max(0,x1-2.47) + 0.334*x3 - 0.302*max(0,x2-1.67) * x4 + 0.291*max(0,x2-1.93) + 0.246*max(0,x2-1.40) * max(0,x2-1.67) + 0.244*max(0,x2-1.67) * log10(x3) + 0.218*max(0,1.40-x2) - 0.0958*max(0,x2-1.40) * x1 + 0.0865*max(0,x1-2.20) + 0.0855*max(0,x1-1.40) - 0.0753*max(0,2.20-x3) - 0.0522*max(0,2.47-x3) + 0.0330*x3^2 + 0.0304*max(0,x2-1.67) + 0.0295*max(0,x1-1.67) + 0.0202*max(0,x2-1.40) + 0.0143*x1^2 - 0.00576*max(0,2.73-x4) - 0.00388*max(0,x2-1.67) * x1 + 0.00111*x4^2
('Number of bases is: ', 56)
('nmse is: ', 2.271972939426071)
model is:
96.7 - 3.76*max(0,x2-2.47) + 3.36*log10(x3) * max(0,x2-1.93) - 3.34*log10(x3) * x1 - 3.19*max(0,x2-2.73) + 2.83*log10(x3) - 2.62*max(0,x1-2.47) * log10(x3) + 2.62*max(0,x2-2.47) * max(0,x2-1.93) + 2.57*log10(x3) * max(0,1.40-x2) + 2.27*max(0,x2-2.20) * max(0,x2-1.93) - 2.18*log10(x3) * x4 - 1.82*max(0,x1-1.40) * max(0,1.40-x2) - 1.75*max(0,x1-1.40) * max(0,x2-1.93) + 1.69*log10(x3) * max(0,x2-2.47) + 1.66*x1 - 1.06*max(0,x2-1.93) * x4 - 0.997*max(0,x2-1.93) * x1 + 0.946*x4 - 0.930*max(0,1.40-x2) * x4 - 0.921*max(0,x1-2.47) * max(0,1.40-x2) - 0.886*max(0,x1-1.40) * max(0,x2-2.47) + 0.870*max(0,x2-1.67) * max(0,x2-1.93) + 0.821*x4 * x1 + 0.821*max(0,x1-1.40) * x4 + 0.812*max(0,x1-2.47) + 0.810*max(0,x1-1.93) - 0.785*max(0,x2-2.47) * x4 - 0.739*max(0,x1-2.47) * max(0,x2-1.67) + 0.715*max(0,x1-2.47) * x4 - 0.597*max(0,x1-2.47) * max(0,x2-2.47) - 0.567*max(0,x2-2.47) * x1 + 0.424*max(0,x2-1.40) * log10(x3) + 0.408*max(0,x2-1.40) * max(0,x2-1.93) + 0.366*max(0,1.40-x2) - 0.325*max(0,x2-1.40) * max(0,x1-2.47) - 0.320*max(0,x2-1.67) * x4 + 0.317*max(0,x2-1.67) + 0.303*x3 - 0.256*max(0,x2-2.20) * x4 + 0.247*max(0,x2-1.67) * log10(x3) + 0.193*max(0,x2-1.40) * max(0,x2-1.67) + 0.145*max(0,x2-1.93) - 0.139*max(0,x2-1.67) * x1 + 0.0886*max(0,x1-2.20) - 0.0788*max(0,x2-1.40) * x1 + 0.0628*max(0,x1-1.40) + 0.0596*max(0,x1-2.73) - 0.0587*max(0,2.47-x3) - 0.0459*max(0,x2-1.40) * x4 + 0.0443*max(0,x2-1.40) - 0.0420*max(0,2.20-x3) + 0.0330*x3^2 + 0.0298*max(0,x1-1.67) + 0.0158*x1^2 + 0.00633*max(0,1.40-x3) - 0.00549*max(0,2.73-x4) + 0.00107*x4^2
('Number of bases is: ', 59)
('nmse is: ', 2.2689292958813225)

model is:
96.6 - 4.55*max(0,x2-2.47) - 3.73*max(0,x2-2.73) - 3.35*log10(x3) * x1 + 3.30*max(0,x2-2.47) * max(0,x2-1.93) + 3.25*log10(x3) * max(0,x2-1.93) + 3.24*log10(x3) + 2.60*log10(x3) * max(0,1.40-x2) - 2.60*max(0,x1-2.47) * log10(x3) + 2.40*max(0,x2-2.20) * max(0,x2-1.93) - 2.20*log10(x3) * x4 + 1.79*log10(x3) * max(0,x2-2.47) - 1.71*max(0,x1-1.40) * max(0,1.40-x2) + 1.68*x1 - 1.65*max(0,x1-1.40) * max(0,x2-1.93) - 0.980*max(0,x2-1.93) * x1 + 0.977*x4 - 0.976*max(0,1.40-x2) * x4 - 0.971*max(0,x1-1.40) * max(0,x2-2.47) - 0.902*max(0,x1-2.47) * max(0,1.40-x2) - 0.897*max(0,x2-1.93) * x4 + 0.886*max(0,x2-1.67) * max(0,x2-1.93) + 0.832*max(0,x1-1.40) * x4 + 0.826*max(0,x1-2.47) + 0.820*max(0,x1-1.93) + 0.811*x4 * x1 - 0.751*max(0,x1-2.47) * max(0,x2-1.67) + 0.712*max(0,x1-2.47) * x4 - 0.668*max(0,x1-2.47) * max(0,x2-2.47) - 0.644*max(0,x2-2.47) * x4 + 0.641*max(0,1.40-x2) + 0.526*max(0,x2-1.67) - 0.522*max(0,x2-2.20) * x4 - 0.490*max(0,x2-2.47) * x1 + 0.482*max(0,x2-1.40) * log10(x3) + 0.445*max(0,x2-1.40) * max(0,x2-1.93) - 0.312*max(0,x2-1.67) * x4 - 0.291*max(0,x2-1.40) * max(0,x1-2.47) + 0.259*x3 + 0.241*max(0,x2-1.67) * log10(x3) - 0.237*max(0,x2-1.67) * x1 + 0.145*max(0,x2-1.40) * max(0,x2-1.67) - 0.140*max(0,1.40-x2) * x1 - 0.0879*max(0,x2-1.40) * x4 + 0.0751*max(0,x1-2.20) - 0.0740*max(0,x2-1.40) * x1 + 0.0695*max(0,x1-1.40) + 0.0673*max(0,1.40-x3) + 0.0600*max(0,x2-1.40) - 0.0592*max(0,1.40-x1) - 0.0520*max(0,2.47-x3) + 0.0451*max(0,x2-1.93) + 0.0396*max(0,x1-2.73) + 0.0371*max(0,x1-1.67) + 0.0309*x3^2 - 0.0173*max(0,2.20-x3) + 0.0100*x1^2 - 0.00512*max(0,2.73-x4) + 0.000954*x4^2 + 0.000292*max(0,x4-1.40)
('Number of bases is: ', 61)
('nmse is: ', 2.267313931033559)
model is:
96.6 - 4.85*max(0,x2-2.47) - 3.90*max(0,x2-2.73) + 3.52*max(0,x2-2.47) * max(0,x2-1.93) + 3.39*log10(x3) - 3.36*log10(x3) * x1 + 3.20*log10(x3) * max(0,x2-1.93) - 2.61*max(0,x1-2.47) * log10(x3) + 2.61*log10(x3) * max(0,1.40-x2) + 2.44*max(0,x2-2.20) * max(0,x2-1.93) - 2.20*log10(x3) * x4 + 1.83*log10(x3) * max(0,x2-2.47) + 1.69*x1 - 1.60*max(0,x1-1.40) * max(0,x2-1.93) - 1.56*max(0,x1-1.40) * max(0,1.40-x2) - 1.02*max(0,x1-1.40) * max(0,x2-2.47) - 0.992*max(0,1.40-x2) * x4 + 0.988*x4 - 0.971*max(0,x2-1.93) * x1 - 0.906*max(0,x1-2.47) * max(0,1.40-x2) + 0.901*max(0,x2-1.67) * max(0,x2-1.93) + 0.879*max(0,1.40-x2) - 0.840*max(0,x2-1.93) * x4 + 0.837*max(0,x1-1.40) * x4 + 0.833*max(0,x1-2.47) + 0.825*max(0,x1-1.93) + 0.807*x4 * x1 - 0.712*max(0,x1-2.47) * max(0,x2-1.67) + 0.711*max(0,x1-2.47) * x4 - 0.699*max(0,x1-2.47) * max(0,x2-2.47) - 0.616*max(0,x2-2.20) * x4 + 0.603*max(0,x2-1.67) - 0.593*max(0,x2-2.47) * x4 + 0.506*max(0,x2-1.40) * log10(x3) + 0.469*max(0,x2-1.40) * max(0,x2-1.93) - 0.443*max(0,x2-2.47) * x1 - 0.309*max(0,x2-1.67) * x4 - 0.287*max(0,1.40-x2) * x1 - 0.280*max(0,x2-1.67) * x1 - 0.274*max(0,x2-1.40) * max(0,x1-2.47) + 0.241*x3 + 0.237*max(0,x2-1.67) * log10(x3) + 0.121*max(0,x2-1.40) * max(0,x2-1.67) - 0.101*max(0,x2-1.40) * x4 - 0.0945*max(0,1.40-x1) + 0.0915*max(0,1.40-x3) - 0.0803*max(0,x2-1.40) * x1 + 0.0703*max(0,x1-1.40) + 0.0696*max(0,x1-2.20) + 0.0604*max(0,x2-1.40) - 0.0572*max(0,x1-2.47) * max(0,x2-1.93) - 0.0489*max(0,2.47-x3) + 0.0392*max(0,x1-1.67) + 0.0300*x3^2 + 0.0279*max(0,x1-2.73) + 0.0229*max(0,x2-1.93) - 0.0201*max(0,1.93-x2) - 0.00811*max(0,2.20-x3) + 0.00715*x1^2 - 0.00496*max(0,2.73-x4) + 0.000897*x4^2 + 0.000450*max(0,x4-1.40)
('Number of bases is: ', 66)
('nmse is: ', 2.26626792962813)
model is:
96.5 - 4.67*max(0,x2-2.47) - 4.25*max(0,x2-2.73) + 3.98*max(0,x2-2.47) * max(0,x2-1.93) + 3.78*log10(x3) - 3.37*log10(x3) * x1 + 3.04*log10(x3) * max(0,x2-1.93) + 2.62*log10(x3) * max(0,1.40-x2) - 2.61*max(0,x1-2.47) * log10(x3) + 2.53*max(0,x2-2.20) * max(0,x2-1.93) - 2.22*log10(x3) * x4 + 1.87*log10(x3) * max(0,x2-

2.47) + 1.81*x1 + 1.61*max(0,1.40-x2) - 1.41*max(0,x1-1.40) * max(0,x2-1.93) - 1.23*max(0,x1-1.40) * max(0,1.40-x2) - 1.17*max(0,x1-1.40) * max(0,x2-2.47) + 1.05*x4 + 1.05*max(0,x2-1.67) * max(0,x2-1.93) - 1.03*max(0,1.40-x2) * x4 + 0.986*max(0,x1-2.47) - 0.939*max(0,x1-2.47) * max(0,1.40-x2) - 0.924*max(0,x2-1.93) * x1 + 0.882*max(0,x1-1.40) * x4 + 0.870*max(0,x1-1.93) - 0.786*max(0,x2-2.20) * x4 + 0.773*x4 * x1 - 0.735*max(0,x2-1.93) * x4 + 0.699*max(0,x2-1.67) + 0.683*max(0,x1-2.47) * x4 + 0.662*max(0,x2-1.40) * max(0,x2-1.93) - 0.614*max(0,1.40-x2) * x1 + 0.588*max(0,x2-1.40) * log10(x3) - 0.521*max(0,x1-2.47) * max(0,x2-2.47) - 0.493*max(0,x2-2.47) * x4 - 0.490*max(0,x1-2.47) * max(0,x2-1.93) - 0.489*max(0,x2-1.40) * max(0,x2-2.47) - 0.466*max(0,x2-1.93) - 0.420*max(0,x2-1.67) * x1 - 0.370*max(0,x2-2.47) * x1 - 0.361*max(0,x2-1.40) * max(0,x1-2.47) - 0.333*max(0,x2-1.67) * max(0,x2-2.47) - 0.310*max(0,x2-1.67) * x4 - 0.276*max(0,x1-2.47) * max(0,x2-1.67) + 0.229*max(0,x2-1.40) * max(0,x2-1.67) - 0.214*max(0,1.93-x2) + 0.201*x3 + 0.168*max(0,x2-1.67) * log10(x3) + 0.153*max(0,1.40-x3) + 0.143*log10(x3) * max(0,x2-2.20) - 0.132*max(0,1.40-x1) - 0.129*max(0,x2-1.40) * x4 + 0.113*max(0,x2-2.20) - 0.111*max(0,x1-2.47) * max(0,x2-2.20) - 0.0970*max(0,x2-1.40) * x1 + 0.0696*max(0,x1-2.20) + 0.0323*max(0,x1-1.40) - 0.0321*max(0,x1-1.40) * x1 - 0.0296*max(0,2.47-x3) + 0.0281*x3^2 + 0.0278*max(0,x2-1.40) + 0.0206*max(0,x1-1.67) - 0.0128*max(0,x1-2.47) * x1 - 0.00457*max(0,2.73-x4) + 0.00109*x1^2 + 0.000825*max(0,x4-1.40) + 0.000759*x4^2
('Number of bases is: ', 73)
('nmse is: ', 2.2661927605111387)
model is:
96.3 - 5.09*max(0,x2-2.73) + 4.33*max(0,x2-2.47) * max(0,x2-1.93) + 4.25*log10(x3) - 4.03*max(0,x2-2.47) - 3.39*log10(x3) * x1 + 2.83*log10(x3) * max(0,x2-1.93) + 2.67*log10(x3) * max(0,1.40-x2) + 2.66*max(0,x2-2.20) * max(0,x2-1.93) - 2.56*max(0,x1-2.47) * log10(x3) + 2.34*max(0,1.40-x2) - 2.22*log10(x3) * x4 + 2.05*x1 + 1.97*log10(x3) * max(0,x2-2.47) - 1.32*max(0,x1-1.40) * max(0,x2-1.93) + 1.28*max(0,x1-2.47) - 1.26*max(0,x1-1.40) * max(0,x2-2.47) + 1.13*max(0,x2-1.67) * max(0,x2-1.93) + 1.08*x4 - 1.06*max(0,1.40-x2) * x4 - 1.06*max(0,x2-1.93) - 1.00*max(0,x1-2.47) * max(0,1.40-x2) - 0.995*max(0,x2-1.67) * max(0,x2-2.47) + 0.977*max(0,x1-1.93) - 0.970*max(0,x2-1.40) * max(0,x2-2.47) - 0.950*max(0,x1-1.40) * max(0,1.40-x2) + 0.906*max(0,x1-1.40) * x4 - 0.865*max(0,1.40-x2) * x1 - 0.861*max(0,x2-1.93) * x1 - 0.800*max(0,x2-2.20) * x4 + 0.785*max(0,x2-1.40) * max(0,x2-1.93) + 0.760*max(0,x2-2.20) + 0.752*x4 * x1 - 0.710*max(0,x2-1.93) * x4 + 0.687*max(0,x1-2.47) * x4 - 0.574*max(0,x1-2.47) * max(0,x2-1.93) + 0.557*max(0,x2-1.40) * log10(x3) - 0.549*max(0,x1-2.47) * max(0,x2-2.47) - 0.524*max(0,1.93-x2) + 0.502*max(0,x2-1.67) - 0.490*max(0,x2-2.47) * x4 - 0.481*max(0,x1-2.73) - 0.457*max(0,x2-1.67) * x1 + 0.389*max(0,x2-1.67) * log10(x3) - 0.376*max(0,x2-1.40) * max(0,x1-2.47) - 0.312*max(0,x2-1.67) * x4 + 0.244*max(0,x2-1.40) * max(0,x2-2.20) - 0.243*max(0,x1-1.67) - 0.236*max(0,x2-2.47) * x1 - 0.232*max(0,x1-2.47) * max(0,x2-1.67) + 0.223*max(0,x2-1.40) * max(0,x2-1.67) + 0.214*max(0,x2-1.67) * max(0,x2-2.20) + 0.175*max(0,1.40-x3) - 0.164*max(0,x2-2.20) * max(0,x2-2.47) - 0.147*max(0,x2-2.20) * x1 - 0.143*max(0,x2-1.40) * x4 + 0.143*x3 - 0.113*max(0,x2-1.40) * x1 + 0.0740*log10(x3) * max(0,x2-2.20) + 0.0577*max(0,1.67-x3) + 0.0439*max(0,x1-2.20) - 0.0264*max(0,x1-2.47) * max(0,x2-2.20) + 0.0259*x3^2 - 0.0249*x1^2 - 0.0197*max(0,x1-2.47) * x1 + 0.0193*max(0,x2-1.40) - 0.0109*max(0,x1-1.40) * x1 - 0.00883*max(0,1.40-x1) - 0.00739*max(0,2.47-x3) - 0.00420*max(0,2.73-x4) + 0.00399*max(0,x1-1.40) + 0.00113*max(0,x4-1.40) - 0.000770*max(0,x1-1.40) * max(0,x2-2.20) + 0.000639*x4^2

Cd_wedge.txt

('Number of bases is: ', 0)

('nmse is: ', 9.142765158861952)
model is:
0.0271
('Number of bases is: ', 1)
('nmse is: ', 8.499975532459482)
model is:
0.0450 - 5.95e-6*x4
('Number of bases is: ', 2)
('nmse is: ', 8.078290478581488)
model is:
0.171 - 0.0356*log10(x4) - 0.00937*log10(x4) * log10(x3)
('Number of bases is: ', 3)
('nmse is: ', 7.814776760089645)
model is:
0.0666 + 0.0437*max(0,5.93-x3) * x1 - 0.0142*log10(x4) + 1.28e-5*max(0,2333-x4)
('Number of bases is: ', 4)
('nmse is: ', 6.1487304384523265)
model is:
0.357 + 0.212*x1 - 0.0989*log10(x3) - 0.0926*log10(x4) + 5.51e-6*x5
('Number of bases is: ', 5)
('nmse is: ', 6.032348516235229)
model is:
0.386 + 0.253*x1 - 0.115*log10(x3) - 0.101*log10(x4) + 0.000330*log10(x5) + 6.40e-6*x5
('Number of bases is: ', 6)
('nmse is: ', 6.020781023961515)
model is:
0.382 - 0.0985*log10(x4) - 0.0825*log10(x3) + 0.0673*log10(x5) * x1 - 0.0211*x2 * log10(x3) + 5.10e-6*x5 + 4.61e-11*x5^2
('Number of bases is: ', 7)
('nmse is: ', 5.811980910848709)
model is:
0.375 - 0.523*log10(x3) * x1 + 0.169*log10(x5) * x1 - 0.104*log10(x4) - 0.0394*log10(x3) - 0.0238*x2 * log10(x3) + 4.18e-6*x5 + 1.09e-10*x5^2
('Number of bases is: ', 8)
('nmse is: ', 4.692854609158447)
model is:
0.618 - 0.952*log10(x3) * x1 + 0.730*log10(x5) * x1 - 0.602*log10(x3) - 0.529*log10(x4) * x1 + 0.177*log10(x4) * log10(x3) - 0.176*log10(x4) - 0.0271*x2 * log10(x3) + 1.30e-10*x5^2
('Number of bases is: ', 11)
('nmse is: ', 4.549350023814631)
model is:
0.707 - 0.992*log10(x3) * x1 + 0.783*log10(x5) * x1 - 0.745*log10(x3) - 0.594*log10(x4) * x1 + 0.220*log10(x4) * log10(x3) - 0.199*log10(x4) + 0.0492*x1 - 0.0274*x2 * log10(x3) + 0.0163*x1^2 - 0.00297*log10(x5) + 1.19e-10*x5^2

('Number of bases is: ', 12)
('nmse is: ', 4.280602987460225)
model is:
0.959 - 1.12*log10(x3) * x1 - 1.07*log10(x3) + 0.930*log10(x5) * x1 - 0.792*log10(x4) * x1 + 0.320*log10(x4) * log10(x3) - 0.249*log10(x4) + 0.222*x1 - 0.0331*x2 * log10(x3) + 0.0296*x2 * x1 - 0.0281*log10(x5) + 0.0230*x1^2 + 1.91e-10*x5^2
('Number of bases is: ', 14)
('nmse is: ', 4.071335680851557)
model is:
1.17 - 4.62*log10(x2) * x1 + 1.86*log10(x2) * log10(x3) + 1.46*x2 * x1 - 1.26*log10(x3) * x1 - 1.04*log10(x4) * x1 + 0.888*log10(x5) * x1 - 0.881*log10(x3) - 0.638*x2 * log10(x3) + 0.438*log10(x4) * log10(x3) - 0.305*log10(x4) - 0.0362*log10(x5) + 0.0217*x1^2 - 0.00130*log10(x2) + 2.71e-10*x5^2
('Number of bases is: ', 15)
('nmse is: ', 4.042709183462439)
model is:
1.29 - 7.47*log10(x2) * x1 + 2.62*log10(x2) * log10(x3) + 2.33*x2 * x1 - 1.33*log10(x3) * x1 - 1.12*log10(x4) * x1 + 0.896*log10(x5) * x1 - 0.834*x2 * log10(x3) - 0.821*log10(x3) - 0.539*x1 - 0.482*log10(x2) + 0.471*log10(x4) * log10(x3) - 0.333*log10(x4) + 0.112*log10(x2) * log10(x4) - 0.0401*log10(x5) + 2.92e-10*x5^2
('Number of bases is: ', 18)
('nmse is: ', 4.02152092138915)
model is:
1.43 - 8.72*log10(x2) * x1 + 2.92*log10(x2) * log10(x3) + 2.73*x2 * x1 - 1.33*log10(x3) * x1 - 1.12*log10(x4) * x1 - 0.940*x1 - 0.912*x2 * log10(x3) + 0.903*log10(x5) * x1 - 0.795*log10(x3) - 0.554*log10(x2) + 0.483*log10(x4) * log10(x3) - 0.350*log10(x4) + 0.149*log10(x2) * log10(x4) - 0.107*x2 - 0.0420*log10(x5) + 0.0250*x2^2 + 0.00290*x2 * log10(x4) + 3.00e-10*x5^2

Heatflux_flatplate.txt

('Number of bases is: ', 0)
('nmse is: ', 19.767544808991673)
model is:
0.00344
('Number of bases is: ', 1)
('nmse is: ', 14.678437218217045)
model is:
0.00626 + 0.000699*log10(x1) * log10(x2)
('Number of bases is: ', 2)
('nmse is: ', 6.675645623137373)
model is:
0.0147 + 0.0319*x1 - 0.00391*log10(x2)
('Number of bases is: ', 3)
('nmse is: ', 6.319695806879151)

model is:
0.0163 + 0.0345*x1 - 0.00440*log10(x2) + 6.97e-5*log10(x1)
('Number of bases is: ', 4)
('nmse is: ', 6.014054150956064)
model is:
0.0237 + 0.0351*x1 - 0.00682*log10(x2) + 8.98e-5*log10(x1) + 2.55e-7*x2
('Number of bases is: ', 5)
('nmse is: ', 4.559636617784359)
model is:
0.00702 + 0.0268*x1^2 + 0.0122*max(0,x1-0.0489) - 0.000742*log10(x2) + 0.000548*log10(x1) * log10(x2) + 1.60e-5*max(0,2800-x2) * x1
('Number of bases is: ', 6)
('nmse is: ', 1.4510256400190196)
model is:
0.000699 + 0.0223*x1 - 0.00299*max(0,0.154-x1) + 1.04e-5*max(0,2800-x2) * x1 + 3.73e-6*max(0,6400-x2) * x1 + 2.79e-6*max(0,2800-x2) * max(0,x1-0.0489) - 3.50e-12*x2^2
('Number of bases is: ', 7)
('nmse is: ', 1.441456390232969)
model is:
0.000698 + 0.0223*x1 - 0.00298*max(0,0.154-x1) + 1.03e-5*max(0,2800-x2) * x1 + 3.73e-6*max(0,6400-x2) * x1 + 2.96e-6*max(0,2800-x2) * max(0,x1-0.0489) - 3.56e-12*x2^2 + 1.49e-12*max(0,5200-x2) * max(0,2800-x2)
('Number of bases is: ', 8)
('nmse is: ', 1.277558686617337)
model is:
0.00794 + 0.172*x1^2 - 0.116*log10(x1) * x1 + 0.0804*x1 - 0.0360*log10(x2) * x1 - 0.00471*log10(x2) + 0.00143*log10(x1) - 0.00130*log10(x1) * log10(x2) + 2.46e-7*x2
('Number of bases is: ', 9)
('nmse is: ', 1.1023542153135928)
model is:
0.0125 + 0.170*x1^2 - 0.117*log10(x1) * x1 + 0.0870*x1 - 0.0377*log10(x2) * x1 - 0.00639*log10(x2) + 0.00204*log10(x1) - 0.00150*log10(x1) * log10(x2) + 6.39e-7*x2 - 2.10e-11*x2^2
('Number of bases is: ', 10)
('nmse is: ', 1.0939993105898445)
model is:
0.000637 + 0.0235*x1 - 0.00211*max(0,0.154-x1) + 1.10e-5*max(0,2800-x2) * x1 + 3.48e-6*max(0,6400-x2) * x1 + 1.49e-6*max(0,2800-x2) * max(0,x1-0.0489) + 4.07e-7*max(0,5200-x2) * max(0,x1-0.0489) - 2.90e-7*max(0,5200-x2) * max(0,0.154-x1) - 1.55e-10*x2 * max(0,2800-x2) + 5.58e-11*max(0,5200-x2) * max(0,2800-x2) - 5.28e-12*x2^2
('Number of bases is: ', 11)
('nmse is: ', 0.9117660084162871)
model is:
0.000585 + 0.0260*x1 - 0.00223*max(0,0.154-x1) + 1.23e-5*max(0,2800-x2) * x1 + 3.02e-6*max(0,6400-x2) * x1 + 7.20e-7*max(0,5200-x2) * max(0,x1-0.0489) - 3.30e-7*x2 * x1 - 5.39e-8*max(0,5200-x2) *

max(0,0.154-x1) - 1.40e-9*x2 * max(0,x1-0.0489) - 2.37e-10*x2 * max(0,2800-x2) + 6.06e-11*max(0,5200-x2) * max(0,2800-x2) - 4.12e-12*x2^2

Heatflux_sphere.txt

('Number of bases is: ', 0)
('nmse is: ', 14.16355573042179)
model is:
9180
('Number of bases is: ', 1)
('nmse is: ', 10.813060845161639)
model is:
1691 - 7.43*x1 * log10(x3)
('Number of bases is: ', 2)
('nmse is: ', 6.761750130831229)
model is:
-7057 + 7406*x1 * x2 - 13.1*x1 * log10(x3)
('Number of bases is: ', 3)
('nmse is: ', 5.33902143074762)
model is:
-1.18e4 + 1.11e4*x1 * x2 - 15.5*x1 * log10(x3) + 0.00129*x1^2
('Number of bases is: ', 4)
('nmse is: ', 5.027829051043651)
model is:
-1.92e4 + 1.22e4*x1 * x2 + 1087*log10(x5) - 16.1*x1 * log10(x3) + 0.00163*x1^2
('Number of bases is: ', 5)
('nmse is: ', 4.735086477174632)
model is:
-2.28e4 + 1.17e4*x1 * x2 + 1989*log10(x5) - 701*log10(x2) * log10(x3) - 19.5*x1 * log10(x3) - 0.0188*x4
('Number of bases is: ', 6)
('nmse is: ', 4.699825564535121)
model is:
-2.32e4 - 5.40e8*x2^2 + 1.25e4*x1 * x2 + 2056*log10(x5) - 706*log10(x2) * log10(x3) - 19.4*x1 * log10(x3) - 0.0202*x4
('Number of bases is: ', 7)
('nmse is: ', 4.518443658651105)
model is:
-2.09e4 - 1.72e9*x2^2 + 1.25e4*x1 * x2 + 2149*log10(x5) - 1214*log10(x2) * log10(x3) - 21.7*x1 * log10(x3) + 0.907*x1 * log10(x2) - 0.0222*x4
('Number of bases is: ', 8)
('nmse is: ', 4.246682694588461)
model is:
-4560 - 8.62e7*x3 * x2 + 1.13e6*x3^2 + 1.99e4*x1 * x2 - 254*x1 * x3 - 17.8*x4 * x2 + 0.0235*x1^2 - 0.0105*x4 + 0.00178*x5

('Number of bases is: ', 9)
('nmse is: ', 3.6453647086756886)
model is:
-5085 - 3.88e9*x2^2 - 1.04e8*x3 * x2 + 1.85e6*x3^2 + 2.72e4*x1 * x2 - 348*x1 * x3 - 29.6*x4 * x2 + 0.0246*x1^2 - 0.0100*x4 + 0.00208*x5
('Number of bases is: ', 10)
('nmse is: ', 3.602762669863214)
model is:
-5293 - 4.18e9*x2^2 - 1.05e8*x3 * x2 + 1.91e6*x3^2 + 2.78e4*x1 * x2 - 356*x1 * x3 - 30.6*x4 * x2 + 0.0247*x1^2 - 0.00997*x4 + 0.00294*x5 - 7.59e-10*x5^2
('Number of bases is: ', 11)
('nmse is: ', 2.7653599981250676)
model is:
-4.91e4 - 3.00e8*x2^2 - 7.68e6*x2 - 4.58e6*log10(x3) * x2 - 1.90e4*log10(x2) + 9598*x1 * x2 - 5794*log10(x2) * log10(x3) + 2433*log10(x5) - 38.6*x1 * log10(x3) + 20.5*x1 * log10(x2) + 0.0294*x1^2 - 0.0283*x4
('Number of bases is: ', 12)
('nmse is: ', 2.7214675867809817)
model is:
-4.72e4 - 7.17e7*x2^2 - 9.08e6*x2 - 5.65e6*log10(x3) * x2 - 1.94e4*log10(x2) + 8944*x1 * x2 - 5315*log10(x2) * log10(x3) + 3007*log10(x3) + 2449*log10(x5) - 39.7*x1 * log10(x3) + 22.1*x1 * log10(x2) + 0.0323*x1^2 - 0.0286*x4
('Number of bases is: ', 14)
('nmse is: ', 2.616452340114885)
model is:
-2.38e4 - 7.57e8*x2^2 - 1.48e7*x2 - 7.96e6*log10(x3) * x2 + 5.81e5*log10(x5) * x2 + 1.97e4*log10(x3) - 1.84e4*log10(x2) + 9769*x1 * x2 - 3842*log10(x2) * log10(x3) - 1611*log10(x5) * log10(x3) - 41.5*x1 * log10(x3) + 22.8*x1 * log10(x2) - 8.70*x1 + 0.0377*x1^2 - 0.0291*x4
('Number of bases is: ', 15)
('nmse is: ', 2.5968822180285995)
model is:
-2.06e4 - 7.18e8*x2^2 - 1.84e7*x2 - 8.57e6*log10(x3) * x2 + 1.03e6*log10(x5) * x2 + 2.32e4*log10(x3) - 1.89e4*log10(x2) + 9952*x1 * x2 - 3430*log10(x2) * log10(x3) - 1868*log10(x5) * log10(x3) + 173*log10(x5) * log10(x2) - 41.9*x1 * log10(x3) + 22.9*x1 * log10(x2) - 10.8*x1 + 0.0390*x1^2 - 0.0292*x4
('Number of bases is: ', 16)
('nmse is: ', 2.572181895911521)
model is:
-6372 - 7.31e8*x2^2 - 2.57e7*x2 - 9.40e6*log10(x3) * x2 + 2.16e6*log10(x5) * x2 + 3.01e4*log10(x3) - 1.81e4*log10(x2) + 9751*x1 * x2 - 2891*log10(x2) * log10(x3) - 2620*log10(x5) * log10(x3) - 2026*log10(x5) + 56.6*log10(x5) * log10(x2) - 42.4*x1 * log10(x3) + 23.6*x1 * log10(x2) - 12.0*x1 + 0.0408*x1^2 - 0.0294*x4

Eureqa

go-eureqa.exe -data="test.txt"

go-eureqa.exe -data="artifical_plus.txt"

Final Results

0: 3 1.503261 (1.0437e+02 + X_0)

1: 4 0.847366 (1.0293e+02 + (X_0)^2)

2: 11 0.690317 (1.0327e+02 + X_0 + { (-1.3047e+00 + 8.8848e-01*X_0*X_3) }/{ 1.4565e+00 })

3: 13 0.612277 (1.0293e+02 + X_0 + { ({ -1.1186e+00 }/{ X_0 } + 8.8848e-01*X_0*X_3) }/{ 1.4565e+00 })

4: 3 1.571222 (1.0409e+02 + X_0)

5: 4 0.915643 (1.0312e+02 + (X_0)^2)

6: 6 0.847366 ((X_0)^2 + 1.0000e+00*1.0293e+02)

7: 11 0.733583 (1.0327e+02 + X_0 + { (-2.5468e-03 + 6.1002e-01*(X_0)^2) }/{ 1.4565e+00 })

8: 13 0.690317 (X_0 + { (-1.3047e+00 + 8.8848e-01*X_0*X_3) }/{ 1.4565e+00 } + 1.0000e+00*1.0327e+02)

9: 15 0.612277 (X_0 + { ({ -1.1186e+00 }/{ X_0 } + 8.8848e-01*X_0*X_3) }/{ 1.4565e+00 } + 1.0000e+00*1.0293e+02)

10: 3 6.995900 (1.1204e+02 + X_3)

11: 4 4.893858 (1.1385e+02 + -(X_0))

12: 5 2.762450 (9.1218e+01 + 6.7252e+00*X_0)

13: 6 1.157325 2.7572e+00*(3.4804e+01 + X_0 + X_3)

14: 3 7.776971 (1.1283e+02 + X_2)

15: 4 5.170128 (1.1422e+02 + -(X_1))

16: 5 2.928180 (1.1558e+02 + { X_1 }/{ -2.4201e-01 })

17: 6 1.363650 2.7755e+00*(3.4228e+01 + X_0 + X_3)

18: 10 1.157325 2.7572e+00*(3.4804e+01 + 1.0000e+00*X_0 + 1.0000e+00*X_3)

19: 3 10.535607 (1.1558e+02 + X_0)

20: 4 5.320290 (1.1434e+02 + -(X_0))

21: 5 2.938035 (1.1577e+02 + { X_1 }/{ -2.4201e-01 })

22: 6 2.080473 (1.2032e+02 + -((1.1964e+01 + X_1)))

23: 7 1.510524 (1.1833e+02 + X_0 + -((1.1813e+01 + X_1)))

24: 8 1.412760 (1.1706e+02 + X_0 + X_3 + -((1.2025e+01 + X_1)))

25: 3 10.535607 (1.1558e+02 + X_1)

26: 4 6.532402 (1.1558e+02 + -(X_2))

27: 5 3.880560 (8.9738e+01 + 6.7252e+00*X_0)

28: 6 2.081379 (1.2042e+02 + -((1.1375e+01 + X_1)))

29: 16 1.794880 ({ 3.6838e+00 }/{ X_0 } + 1.0000e+00*8.6837e+01 + 6.7252e+00*X_0 + 6.7185e-01*X_1 + 1.0000e+00*X_3)

30: 21 1.534501 (8.6837e+01 + { 3.6838e+00 }/{ X_0 } + { 3.6838e+00 }/{ X_3 } + { -9.0181e-

01*X_1*cos(X_3) }/{ X_2 } + 6.7252e+00*X_0 + 1.0000e+00*X_3)
31: 3 23.701550 (8.1347e+01 + X_1)

go-eureqa.exe -data="artifical_plus2.txt"
Final Results

0: 3 11.122978 -((X_1)^3)
1: 4 9.968431 -1.1921e+00*(X_1)^3
2: 5 7.059069 -6.0528e-01*(-(X_1))^4
3: 7 5.704560 ((X_3)^2 + -1.9159e+00*(X_1)^3)
4: 8 5.329786 { (X_3 + -4.1046e-01*(X_1)^4) }/{ 6.1426e-01 }
5: 9 4.775733 { (9.4172e-01 + X_3 + { (X_1)^4 }/{ -2.6896e+00 }) }/{ 5.7413e-01 }
6: 10 4.166238 { (4.1046e-01 + X_0 + X_3 + -4.1046e-01*(X_1)^4) }/{ 6.1426e-01 }
7: 11 3.833625 { (X_0 + X_3 + cos(X_1) + -4.1046e-01*(X_1)^4) }/{ 6.1426e-01 }
8: 3 11.122978 -((X_1)^3)
9: 4 10.518023 { -6.6022e+00 }/{ sin(X_1) }
10: 5 7.637722 -6.6822e-01*1.0000e+00*(X_1)^4
11: 6 7.491609 (X_0 + -1.7923e+00*(X_1)^3)
12: 7 5.892981 (4.3772e+00 + X_0 + -2.1259e+00*(X_1)^3)
13: 8 5.755457 X_1*cos(X_1)*(6.3914e+00 + (X_1)^2)
14: 11 5.604703 cos(X_1)*(2.0407e+00 + X_1)*(sin(X_0) + (X_1)^2)
15: 12 5.496194 { (X_1 + 6.9752e+00*sin((1.2114e+00 + X_1))*(X_1)^3) }/{ X_1 }
16: 13 5.454139 cos(X_1)*(2.0407e+00 + X_1)*((X_1)^2 + 1.0000e+00*sin(X_3))
17: 15 5.401366 6.2459e-01*(X_1)^2*(3.1132e+00 + X_3 + -((X_1)^2) + cos(X_1) + { X_1 }/{ -9.3702e-01 })
18: 16 5.146235 cos(X_1)*(2.0407e+00 + X_1)*(7.0442e-01 + sin(-1.9412e+00*(-2.2777e+00 + X_3)) + (X_1)^2)
19: 21 5.141369 cos(X_1)*(3.1837e-01 + sin(-3.8407e-01*(-2.2777e+00 + X_3)) + (X_1)^2)*(2.0407e+00 + X_1 + -3.8407e-01*(-2.2777e+00 + X_3))
20: 22 5.064263 cos(X_1)*(7.0442e-01 + sin(-3.8407e-01*(-2.2777e+00 + X_0)) + (X_1)^2)*(2.0407e+00 + X_1 + sin(-3.8407e-01*(-2.2777e+00 + X_3)))
21: 23 4.449073 6.2459e-01*(X_1)^2*(-((X_1)^2) + { (2.6546e+00 + sin(X_1) + cos(X_1) + { X_1 }/{ -9.3702e-01 } + 1.0000e+00*X_0*X_3) }/{ X_1 })
22: 3 11.122978 -((X_1)^3)
23: 4 10.537437 -1.0928e+00*(X_1)^3
24: 5 8.629967 { -((X_1)^4) }/{ X_3 }
25: 6 7.520618 (X_0 + -1.6441e+00*(X_1)^3)
26: 7 7.154285 -6.2459e-01*(X_1)^2*(-(X_1))^2
27: 8 6.730952 cos(X_1)*(X_1)^2*(3.2220e+00 + X_2)
28: 10 6.607990 6.2459e-01*(X_1)^2*(-((X_1)^2) + sin(X_1))
29: 12 5.893958 (2.1028e+00 + X_2 + (X_3)^2 + -1.0000e+00*(X_1)^2*(2.1028e+00 + X_1))
30: 14 5.677488 (-1.0000e+00 + X_2 + -1.0000e+00*(X_1)^2*(X_1 + X_2) + X_0*X_2*X_3)
31: 22 5.243866 6.2459e-01*(X_1)^2*(-((X_1)^2) + { (2.6546e+00 + sin(X_1) + cos(X_1) + { X_1 }/{ -9.3702e-01 } + 1.0000e+00*X_3) }/{ X_1 })

go-eureqa.exe -data="artifical_times.txt"

Final Results

0: 3 2.284771 (1.0056e+02 + X_0)
1: 4 2.012787 (9.9654e+01 + (X_0)^2)
2: 5 1.756972 (9.9381e+01 + X_0*X_3)
3: 8 1.717916 1.7414e+00*(5.7695e+01 + X_0*sin(X_1))
4: 9 1.121218 1.7414e+00*(5.6877e+01 + X_0*X_3*sin(X_1))
5: 3 2.289405 (1.0046e+02 + X_0)
6: 4 2.017065 (9.9193e+01 + (X_0)^2)
7: 5 1.773293 (9.9654e+01 + X_0*X_3)
8: 7 1.756972 (1.0000e+00*9.9381e+01 + X_0*X_3)
9: 3 2.289801 (1.0046e+02 + X_0)
10: 4 2.020498 (9.9718e+01 + (X_0)^2)
11: 5 1.783062 (9.9718e+01 + X_0*X_3)
12: 3 2.294270 (1.0040e+02 + X_0)
13: 4 2.029976 (9.9780e+01 + (X_0)^2)
14: 5 1.784627 (9.7299e+01 + X_3 + (X_0)^2)
15: 3 2.302041 (1.0099e+02 + X_0)
16: 4 2.031369 (9.9789e+01 + (X_0)^2)
17: 5 1.794410 (9.9780e+01 + X_0*X_3)
18: 7 1.784627 (9.7299e+01 + (X_0)^2 + 1.0000e+00*X_3)
19: 3 2.309820 (1.0027e+02 + X_0)
20: 4 2.051447 (9.9887e+01 + (X_0)^2)
21: 5 1.810199 (9.7159e+01 + X_3 + (X_0)^2)
22: 3 20.638199 (8.1115e+01 + X_0)
23: 4 10.834665 ((8.2709e+00 + X_0))^2
24: 5 2.119196 1.7414e+00*(5.6877e+01 + X_0)
25: 7 1.868870 1.7414e+00*(5.5576e+01 + X_0*X_3)
26: 3 21.575252 (8.0178e+01 + X_2)
27: 4 11.850665 ((8.2709e+00 + X_3))^2
28: 5 2.271687 1.7414e+00*(5.7695e+01 + X_0)
29: 7 2.070970 1.7414e+00*(5.7695e+01 + { X_0 }/{ X_1 })
30: 3 25.451825 (7.6302e+01 + X_0)
31: 4 12.657213 (8.0964e+01 + (X_0)^3)

go-eureqa.exe -data=" Cd_wedge.txt"

Final Results

0: 3 0.021890 { 3.5572e+01 }/{ X_3 }
1: 5 0.021776 { 3.4148e+01 }/{ (-8.2558e+01 + X_3) }

2: 6 0.020635 { X_4*(X_0)^2 }/{ X_3 }
3: 7 0.017634 { X_0*X_4 }/{ 1.9040e+01*X_3 }
4: 9 0.015717 { X_0*X_4 }/{ 9.3870e-01*X_3*(X_2)^2 }
5: 10 0.009176 { X_0*X_1*X_4 }/{ 5.5131e-01*X_3*(X_2)^2 }
6: 3 0.021890 { 3.5572e+01 }/{ X_3 }
7: 5 0.021777 { 3.3811e+01 }/{ (-8.2558e+01 + X_3) }
8: 7 0.021322 { 4.9100e+01 }/{ (X_3 + { 3.8745e+01 }/{ X_0 }) }
9: 8 0.018859 { X_0*X_4 }/{ X_1*X_2*X_3 }
10: 10 0.016306 { X_0*X_4 }/{ X_3*cos(X_0)*(X_2)^2 }
11: 11 0.015472 sin({ 3.7684e+01*X_4 }/{ (X_3)^2*(-1.4463e+00 + X_2) })
12: 12 0.012165 { X_0*(X_4)^2 }/{ (X_1)^2*(X_2)^2*(X_3)^2 }
13: 3 0.021903 { 3.3805e+01 }/{ X_3 }
14: 5 0.021854 { 3.3160e+01 }/{ (-3.5260e+01 + X_3) }
15: 8 0.020904 { 3.2909e+01 }/{ (X_3 + -2.4300e+00*X_0*X_3) }
16: 9 0.020784 { 3.2909e+01 }/{ (-8.2558e+01 + X_3 + -2.4300e+00*X_0*X_3) }
17: 10 0.020656 { X_0*(2.7389e+00 + X_0 + sin(X_3)) }/{ (X_2)^2 }
18: 11 0.020353 { X_0*(1.9539e+00 + X_0 + X_1 + sin(X_3)) }/{ (X_2)^2 }
19: 12 0.019615 { X_0*(1.0000e+00 + X_1 + 2.2410e+00*sin(X_3)) }/{ (X_2)^2 }
20: 13 0.015390 sin({ 3.8745e+01*X_4 }/{ X_3*(-3.5565e+01 + X_3)*(-1.6749e+00 + X_2) })
21: 16 0.014520 sin({ 3.7684e+01*X_4 }/{ X_3*(-1.4463e+00 + X_2)*(X_2 + X_3 + { 3.8745e+01 }/{ X_0 }) })
22: 3 0.021908 { 3.7684e+01 }/{ X_3 }
23: 5 0.021884 { (3.6485e+01 + X_0) }/{ X_3 }
24: 8 0.020907 { 3.2956e+01 }/{ (X_3 + -2.4300e+00*X_0*X_3) }
25: 11 0.020619 { 3.2956e+01 }/{ (X_3 + { X_2 }/{ X_0 } + -2.4300e+00*X_0*X_3) }
26: 12 0.020304 { 3.3536e+00*((2.6843e+00 + sin(X_3) + 4.4842e+00*X_0))^2 }/{ X_3 }
27: 14 0.017948 { X_0*(1.9539e+00 + X_1 + cos(X_3) + 2.2410e+00*sin(X_3)) }/{ (X_2)^2 }
28: 15 0.017251 { X_0*(1.9539e+00 + sin(X_3))*(8.8884e-01 + X_0 + X_1 + sin(X_4)) }/{ (X_2)^2 }
29: 17 0.016899 { X_0*(X_0 + X_1 + sin(X_4) + 1.0000e+00*1.9539e+00 + 2.2410e+00*sin(X_3)) }/{ (X_2)^2 }
30: 3 0.021912 { 3.7909e+01 }/{ X_3 }
31: 5 0.021897 { 3.8454e+01 }/{ (-3.5260e+01 + X_3) }

go-eureqa.exe -data=" Heatflux_flatplate.txt"
Final Results

0: 3 0.001752 { 1.0590e+01 }/{ X_1 }
1: 4 0.001363 2.2567e-01*(X_0)^2
2: 5 0.001358 { (sin(X_0))^2 }/{ 4.1948e+00 }
3: 6 0.001206 1.6819e-01*((2.0706e-02 + X_0))^2
4: 7 0.001186 { 1.0227e+01 }/{ { 8.6406e-02*X_1 }/{ X_0 } }
5: 9 0.001162 { 8.8479e+00 }/{ (1.2293e+01 + { 8.6406e-02*X_1 }/{ X_0 }) }

6: 10 0.001051 { 1.0227e+01 }/{ (1.0409e+02 + X_0 + { 8.6406e-02*X_1 }/{ X_0 }) }
7: 11 0.001049 { 1.0227e+01 }/{ (1.0409e+02 + { 8.6406e-02*(2.5244e+00 + X_1) }/{ X_0 }) }
8: 12 0.001047 { 1.0859e+01 }/{ (1.2293e+01 + { 1.1084e+01 }/{ X_0 } + { 8.6406e-02*X_1 }/{ X_0 }) }
9: 3 0.001752 { 1.0394e+01 }/{ X_1 }
10: 4 0.001569 1.6819e-01*(X_0)^2
11: 6 0.001351 { (X_0)^2 }/{ (4.2537e+00 + X_0) }
12: 7 0.001202 { 1.0852e+01 }/{ { 8.6406e-02*X_1 }/{ X_0 } }
13: 9 0.001192 { (X_0)^2 }/{ (3.7452e+00 + X_0 + sin((X_1)^2)) }
14: 10 0.001114 ((2.0706e-02 + X_0))^2*(1.6819e-01 + { 3.1741e-03 }/{ X_0 })
15: 3 0.001753 { 1.0335e+01 }/{ X_1 }
16: 5 0.001575 1.6819e-01*(sin(X_0))^2
17: 6 0.001358 { (X_0)^2 }/{ (4.3693e+00 + X_0) }
18: 7 0.001202 1.6819e-01*((2.0875e-02 + sin(X_0)))^2
19: 11 0.001162 { (sin(X_0))^2 }/{ ((1.4427e+00 + 5.2138e+00*sin(X_0)))^2 }
20: 3 0.001753 { 1.0324e+01 }/{ X_1 }
21: 5 0.001728 { 1.0227e+01 }/{ (1.0409e+02 + X_1) }
22: 6 0.001391 1.7431e-01*({ X_0 }/{ -9.1750e-01 })^2
23: 7 0.001349 { (sin(X_0))^2 }/{ (4.2537e+00 + X_0) }
24: 8 0.001205 { (X_0)^2 }/{ (4.1948e+00 + sin((X_1)^2)) }
25: 24 0.001205 { 5.8177e+00 }/{ ({ X_1 }/{ 2.4915e+00 } + { X_1 }/{ (-3.9618e-01 + 7.0506e-02*X_0*X_1) } + { (-3.9618e-01 + X_1) }/{ (-3.9618e-01 + 7.0506e-02*X_0*X_1) }) }
26: 27 0.001200 { 7.0250e+00 }/{ ({ X_1 }/{ 2.2360e+00 } + { X_1 }/{ (4.0061e-01 + 7.0506e-02*X_0*X_1) } + { (7.4325e-01 + X_1 + { X_1 }/{ 1.7866e+00 }) }/{ (4.6822e-01 + 7.0506e-02*X_0*X_1) }) }
27: 29 0.001181 { 6.5284e+00 }/{ ({ X_1 }/{ 2.4915e+00 } + { (3.7331e-01 + X_1) }/{ (4.0061e-01 + 7.0506e-02*X_0*X_1) } + { (7.4325e-01 + X_1 + { X_1 }/{ 2.4915e+00 }) }/{ (4.6822e-01 + 7.0506e-02*X_0*X_1) }) }
28: 3 0.001783 { 9.0232e+00 }/{ X_1 }
29: 5 0.001728 { 1.0227e+01 }/{ (1.0409e+02 + X_1) }
30: 6 0.001462 { (X_0)^2 }/{ (3.2912e+00 + X_0) }
31: 7 0.001350 { (X_0)^2 }/{ (4.1052e+00 + sin(X_0)) }

go-eureqa.exe -data=" Heatflux_sphere.txt"

Final Results

0: 3 4543.927705 1.0157e+01*X_0
1: 5 4469.408135 1.0356e+01*(-1.0184e+02 + X_0)
2: 7 4403.549527 (X_0 + 1.0557e+01*(-1.0319e+02 + X_0))
3: 8 3952.401317 -4.0012e+00*X_0*(-1.8272e+00 + sin(-(X_0)))
4: 10 3941.931909 (-4.0012e+00 + -4.0460e+00*X_0*(-1.9384e+00 + sin(-(X_0))))
5: 13 3921.971386 (1.3488e+00 + { 2.7711e+00 }/{ X_2 } + -4.0012e+00*X_0*(-1.9384e+00 + sin(-(X_0))))
6: 14 3871.724753 (1.3488e+00 + { 9.7128e+00 }/{ sin(X_2) } + -4.0012e+00*X_0*(-1.9384e+00 + sin(-(X_0))))

7: 15 3656.257066 ({ 1.0000e+00 }/{ (X_2)^2 } + 5.9376e+00*X_0 + 5.9376e+00*X_0*X_1*(1.0447e+00 + X_0))
8: 18 3616.731019 (1.0000e+00 + sin(X_4) + { 1.0000e+00 }/{ (X_2)^2 } + 6.8890e+00*X_0 + 6.4940e+00*X_0*X_1*(X_0 + X_1))
9: 20 3592.092682 (1.0000e+00 + sin(X_1) + { 1.0000e+00 }/{ (X_2)^2 } + 6.8890e+00*X_0 + 6.8890e+00*X_0*X_1*(X_0 + { X_0 }/{ X_3 }))
10: 3 4545.191026 1.0356e+01*X_0
11: 5 4531.397459 1.0157e+01*(-1.0157e+01 + X_0)
12: 6 4520.142841 -3.5760e+00*X_0*(-2.9928e+00 + X_2)
13: 7 4415.781632 (1.0356e+01 + 1.1021e+01*(-1.0184e+02 + X_0))
14: 8 4351.704680 (-3.7035e+02 + X_0 + 1.1021e+01*(-1.0319e+02 + X_0))
15: 9 4158.945018 (-2.8239e+00 + 1.2144e+01*X_0 + -7.7099e-01*X_2*X_3)
16: 11 4082.458639 (-2.9382e+00 + -((X_2 + X_2*X_3)) + { X_0 }/{ 8.2345e-02 })
17: 12 4073.574550 (-2.9382e+00 + X_1 + -((1.0000e+00 + X_2*X_3)) + 1.2478e+01*X_0)
18: 13 4069.209599 (-2.9382e+00 + -((8.2345e-02 + X_2*(X_0 + X_3))) + 1.2478e+01*X_0)
19: 14 4013.694973 (1.0000e+00 + 5.9376e+00*X_0 + 6.4940e+00*X_0*X_1*(X_0 + { X_0 }/{ X_3 }))
20: 3 4696.388699 1.2144e+01*X_0
21: 5 4544.088106 (X_0 + { X_0 }/{ 1.0810e-01 })
22: 6 4540.316377 (-2.7581e+01 + X_0 + { X_0 }/{ 1.0810e-01 })
23: 7 4497.160596 (X_0 + { (-4.7738e+01 + X_0) }/{ 1.0810e-01 })
24: 8 4403.549485 (X_0 + X_1 + 1.0557e+01*(-1.0319e+02 + X_0))
25: 11 4360.515395 1.0157e+01*(X_0 + { (X_0 + -3.1354e+00*X_3) }/{ X_0 })
26: 12 4101.053887 (1.0000e+00 + 5.9376e+00*X_0 + 5.9376e+00*X_0*X_1*(1.0447e+00 + X_0))
27: 14 4068.970768 (-2.9382e+00 + X_1 + -((2.2595e+00 + X_2*(X_0 + X_3))) + 1.2478e+01*X_0)
28: 15 4065.862248 (-3.6616e+00 + -(({ -9.1864e-01 }/{ X_2 } + X_2*(X_0 + X_3))) + 1.2478e+01*X_0)
29: 3 5028.497173 6.8890e+00*X_0
30: 4 4830.134063 1.0000e+00*7.6315e+00*X_0
31: 5 4544.884759 (X_2 + 1.0033e+01*X_0)

gplearn

gplearn_paper.py

Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:22:17) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\rochant\Desktop\SymbolicRegressionAlgorithms\gplearn\gplearn_paper.py

artifical_plus.txt

	Population Average		Best Individual			
Gen	Length	Fitness	Length	Fitness	OOB Fitness	Time Left
0	31.03	1601.1143966917869	31	31.900898940340646	32.29183084923351	55.04s
1	33.89	106.90622793550976	39	27.87540144350247	26.48544306871639	55.78s
2	43.44	127.27137042722262	13	9.751334217154705	9.84320310943285	57.09s
3	41.47	89.94630714629533	13	9.812722117510576	9.295584061813802	55.45s
4	35.86	91.80406340377107	15	2.926692445872081	2.637847063453318	52.03s
5	36.97	112.10988634076118	19	1.8517640294113817	1.694074941066577	49.03s
6	36.37	145.17703097404052	13	1.6775461138754926	1.9448137245458188	45.83s
7	19.1	175.24441912533618	27	1.5340038896316475	1.601066506180889	40.97s
8	17.02	612.650391771854	27	0.9803628938982994	1.0032453949824334	36.39s
9	16.95	167.25779272640483	27	0.9938120249494451	0.8832706068753877	32.25s
10	17.09	156.86719254506755	27	0.9629292021169357	1.1587649946828549	28.35s
11	17.43	150.31456682942226	27	0.7105118964130313	0.6921579709828363	24.72s
12	20.58	120.97477912837239	27	0.7071862900951379	0.7218244908345195	21.43s
13	24.14	86.02867474806737	27	0.6132600182033594	0.7231216591932784	18.35s
14	26.41	72.91625634625393	27	0.618672961169293	0.6748347711479651	15.27s
15	27.35	62.83000871674856	27	0.5591176828454824	0.6361948286470369	12.24s
16	27.12	61.897147176674764	27	0.5659734651054769	0.5750368980102599	9.18s
17	27.16	69.89193109937275	31	0.5249803370433979	0.5751832530416001	6.13s
18	27.04	57.31276664661897	31	0.4785849996384913	0.6356630939852865	3.07s
19	27.18	58.35679912777584	33	0.48195946337218337	0.4873817635168119	0.00s

sub(sub(mul(0.140, X2), add(add(X2, div(X2, mul(X3, X3))), sub(div(add(X1, div(X2, X3)), X2), sub(mul(X0, X0), X0)))), div(add(X3, X3), mul(-0.018, X3)))

artifical_plus2.txt

	Population Average		Best Individual			
Gen	Length	Fitness	Length	Fitness	OOB Fitness	Time Left
0	31.03	1521.0878091557802	63	10.469823160809323	12.580155482863502	53.50s
1	23.2	22.418625639852433	9	8.29944916370107	8.443048095238094	49.27s
2	31.31	28.621308264621206	17	5.014735565049373	4.631485703977409	48.57s
3	29.28	27.858663766740392	21	2.4727943517108044	2.223115546937173	46.24s
4	24.3	28.84244967641777	21	2.44671106438751	2.455795030678309	42.51s
5	20.28	27.04239631230495	33	2.054523982206405	2.078740317460318	38.47s
6	20.04	22.673045781041044	27	1.2097030391459074	1.217044603174603	35.24s
7	24.37	20.868590195018424	27	1.2168710640569393	1.15310126984127	32.29s
8	27.36	23.30133396258285	27	1.1737446583629891	1.5378161904761904	30.02s
9	26.27	33.16854908225364	27	1.1752563861209964	1.524330619047619	27.20s
10	25.66	20.075828838582915	23	1.080238040925267	1.1983390317460318	24.33s

11	30.59	19.66926238770216	23	1.081889747330961	1.183604761904762	21.67s
12	32.87	32.7538378944892	23	1.065654017793594	1.3284377777777776	19.02s
13	29.46	22.445640139294714	27	0.9106076298932384	0.9393471428571425	16.24s
14	26.2	17.778675595228602	27	0.8992252829181494	1.0408849047619049	13.42s
15	24.26	17.19453800319043	27	0.893498181494662	1.0919742857142856	10.63s
16	23.69	17.659427289182442	29	0.8710426151620936	0.98215881441984	7.91s
17	24.05	34.85691260232086	23	0.8461286085409252	0.7400619047619048	5.23s
18	24.55	17.529731615424645	23	0.8202371850533808	0.9710298412698412	2.61s
19	23.77	16.47545494690101	23	0.8173021156583631	0.9972125238095236	0.00s

add(sub(add(mul(X2, X0), mul(add(X3, X0), X3)), mul(X1, X1)), sub(X0, mul(add(X1, X1), mul(X1, X1))))

artifical_times.txt

	Population Average		Best Individual			
Gen	Length	Fitness	Length	Fitness	OOB Fitness	Time Left
0	31.03	1597.973965103748	31	29.73165460750754	29.637105410249145	52.65s
1	33.66	104.08576859279403	39	26.74799886409086	24.874014497287813	52.54s
2	42.99	123.68730686085155	13	13.03983481143413	13.472091998321739	54.44s
3	41.13	88.00025602447421	13	13.17780634168324	12.241298347528089	52.89s
4	34.48	77.09066011415032	23	5.511025011297956	5.628116618721433	49.49s
5	40.92	90.68925968256488	21	3.2286373466184215	3.367744646975829	47.68s
6	52.44	101.5550446877442	11	3.1370696805424623	2.722213196537741	46.27s
7	24.32	162.36959285394505	19	2.6972963050310392	2.6760497116573947	41.61s
8	23.9	134.13415535101237	17	2.306722964103045	2.8586664804479116	37.42s
9	23.04	136.82416413345365	67	2.1778088617705973	2.3949524255794366	33.49s
10	24.22	122.14743815383031	85	2.1584594082009727	2.617407156756565	29.76s
11	23.09	135.7888859005279	65	2.0879576824165973	2.269350536762043	26.14s
12	21.27	143.79145957226552	75	2.0122745538660824	2.222243786208399	22.59s
13	19.19	124.9652196911986	19	2.0063507820587687	1.6013794986977257	19.08s
14	18.12	123.25498796891475	19	1.9185378630499217	2.3847264904909324	15.69s
15	18.99	109.64171888253661	23	1.7018018968342778	1.5954259778990247	12.41s
16	19.96	105.01915727719182	35	1.5931149530038569	1.9064535185487048	9.22s
17	20.58	113.91498519579363	35	1.5976833278285059	1.8657007145573925	6.10s
18	21.73	83.56165802420908	35	1.4212939210972073	1.6138277520170408	3.04s
19	23.36	87.2894131603523	35	1.4238328119563943	1.5911792335588952	0.00s

sub(sub(add(add(sub(sub(X0, 0.774), add(X3, X0)), sub(sub(mul(X3, X0), add(X2, 0.710)), 0.774)), sub(X0, 0.774)), sub(add(X1, X1), 0.432)), div(add(X3, X3), mul(-0.018, X3)))

artifical_Cd_wedge.txt

	Population Average		Best Individual			
Gen	Length	Fitness	Length	Fitness	OOB Fitness	Time Left
0	29.43	4.375820071535108e+28	127	0.010719128448091482	0.012075854505530474	1.74m

1 9.06 198760856.7570989 15 0.017988023832180344 0.018397433883293002 1.35m
2 3.47 288003914395143.2 7 0.02260026704063492 0.022848529142857142 1.11m
3 3.06 21335466821955.66 5 0.0227336412318579 0.02686563934998644 57.70s
4 1.73 78967519.03677294 3 0.02323729053079365 0.028724237518095237 50.66s
5 1.5 2.1811835271433e+16 3 0.023638556262962965 0.025112845928571426 45.04s
6 1.45 39580951.06375474 7 0.02402654118116402 0.021620981664761903 40.25s
7 1.46 2.2417717538960528e+17 3 0.02372558644222222 0.0243295743152381 36.16s
8 1.56 2.6372187531815246e+26 5 0.023010294057853514 0.02437576391602593 32.51s
9 1.49 148231713759163.7 3 0.02432580409505467 0.026518751184528 29.10s
10 1.39 389073275.32989043 3 0.023555083812169312 0.025864097985714287 25.80s
11 1.52 163569539.23097983 7 0.025006646610819284 0.02651054679576165 22.69s
12 1.5 153847845606.8845 3 0.024214145917791776 0.027523674779894 19.64s
13 1.47 613847508.7308334 3 0.023432776784444442 0.026964861235238093 16.67s
14 1.57 637998585496862.5 5 0.023006043058201402 0.024414022912894937 13.81s
15 1.45 1832945446542.7407 15 0.019322801648250256 0.01999334374133749 11.00s
16 1.47 66180495689.87054 3 0.024456128914166445 0.025345827812522 8.19s
17 1.48 208642295.72529942 3 0.024269000534011778 0.027029983233913996 5.42s
18 1.48 105665584.00804216 9 0.019633757654336614 0.01681563332493183 2.70s
19 1.45 367878026013.9694 3 0.023822565571851848 0.02345676214857143 0.00s

div(X0, X2)

artifical_Heatflux_flatplate.txt

| Population Average | Best Individual |
---- ---------- ---------- ---------- ---------- ----------
Gen Length Fitness Length Fitness OOB Fitness Time Left
0 38.13 1.222681990406716e+29 15 0.000972268415255915 0.0011713905570498548 55.61s

div(sub(sub(0.808, X0), mul(X1, X0)), mul(add(-0.770, -0.378), div(X1, 0.034)))

artifical_Heatflux_sphere.txt

| Population Average | Best Individual |
---- ---------- ---------- ---------- ---------- ----------
Gen Length Fitness Length Fitness OOB Fitness Time Left
0 29.43 2.5247318035185896e+41 7 3984.832916255144 4211.215697777778 6.41m

1	17.35	5053732907261.99	9	3802.26182553204	3847.5624431746023	5.24m
2	27.6	6.6473270352923896e+16	65	3734.0144414677616	3726.366827114342	5.11m
3	41.25	5.3539726795797535e+22	13	2634.324301125841	2758.9283526080903	5.28m
4	25.65	3.5841453244781437e+19	17	2640.547396352774	2703.1149678581132	4.85m
5	37.78	177712769025208.16	71	2622.417394393974	2666.881833974254	4.64m
6	59.01	806118436591084.1	67	2384.3786305150147	2397.504834499385	4.65m
7	47.96	6.135877418534893e+21	69	2328.428119532658	2305.2846349015804	4.42m
8	37.43	7.702385736555299e+24	61	2279.9413638939373	2286.1224060829263	4.08m
9	54.66	60520653627877.695	81	1886.4854568799778	1917.233685148318	3.81m
10	71.33	5147593235307.232	81	1882.5597553609398	1952.5649988196617	3.58m
11	75.26	142675223187442.7	83	1876.9060782777901	2003.8331570560701	3.32m
12	86.27	43032525215091.55	99	1846.997317168193	1942.4196254118044	3.04m
13	99.64	1.9907856257622584e+18	87	1839.5434805320954	1833.1728424720038	2.74m
14	104.96	3.257804793129515e+31	151	1768.9003082928973	1746.7669058341608	2.39m
15	110.73	4.676862288867244e+16	151	1761.87614534818	1809.9843723366187	1.99m
16	109.03	733292309002361.8	155	1751.7767887959528	1793.9168904469802	1.55m
17	113.2	4.469972699520196e+16	163	1730.6315198921093	1762.0472342093835	1.07m
18	138.0	1.7338060560783094e+17	225	1708.2754469421918	1676.3472323222377	33.35s
19	150.38	2.2218210282237647e+18	223	1698.6107017045726	1763.3296247509684	0.00s

div(add(mul(sub(add(add(mul(mul(X0, X0), div(X1, X2)), X0), X0), mul(X1, X4)), add(sub(0.785, X3), add(X1, X0))), sub(div(sub(X3, X0), mul(sub(add(mul(mul(X0, X0), div(X1, X2)), X0), mul(X1, X3)), add(mul(X1, X3), -0.539))), mul(mul(X0, X0), mul(add(div(add(mul(sub(add(add(div(add(mul(sub(add(mul(mul(X0, X0), div(X1, X2)), X0), mul(X1, X3)), add(sub(0.785, X3), add(X1, X0))), sub(sub(div(X2, X4), div(div(X1, X2), X0)), mul(mul(X0, X0), add(X2, X2)))), sub(add(add(sub(X2, X3), mul(X1, X3)), div(X2, X2)), div(mul(div(X0, X3), mul(X0, X0)), mul(mul(0.108, X2), mul(0.108, X2))))), X0), X0), mul(X1, X3)), sub(0.021, X3)), sub(div(X1, X3), mul(mul(X0, X0), add(X2, X2)))), sub(add(add(sub(X2, X3), sub(-0.627, X1)), div(X2, X2)), sub(add(mul(mul(X0, X0), div(X1, X2)), X0), sub(add(mul(mul(mul(X0, X0), div(X1, X2)), div(X1, X2)), X0), mul(-0.046, X0))))), X0), mul(0.108, X2))))), sub(add(add(sub(X2, X3), sub(-0.627, X1)), div(X2, X2)), div(mul(add(X3, -0.473), mul(X3, X1)), mul(add(mul(mul(X0, X0), add(X2, X2)), X0), mul(0.108, X2)))))

>>>

GPOLS

GPOLS_paper.m

artifical_plus.csv

fitness: 0.958992, mse: 126.280720
 0.525791 * (((x4)+(x1))*(x1)) +
 102.575452

artifical_plus2.csv

fitness: 0.917117, mse: 24741.588277
 -7.526976 * ((x2)*(x2)) +
 21.774349

artifical_times.csv

fitness: 0.786810, mse: 1705.589089
 4.279160 * (x1) +
 1.125629 * ((((x4)/(x3))-(x2))*(x1)) +
 97.083056

Cd_wedge.csv

fitness: 0.682984, mse: 3.700585
 0.201815 * ((((x3)*((x5)*(x3)))*((x1)^(x3)))/((x4)*(x1))) +
 0.011800

Heatflux_flatplate.csv

fitness: 0.909251, mse: 0.000043
 31.601465 * (((((x2)*(x1))^(x1))*(x1))/(x2)) +
 0.001558

Heatflux_sphere.csv

fitness: 0.777709, mse: 418323430297.922974
 25.220063 * (x1) +
 13650.587647 * ((x2)*(x1)) +
 -114069.748374 * (x3) +
 -9092.076308

GPTips2

gp.genes.max_genes = 4;

artifical_plus.csv
mfun =

1.0*x4 - 3.63*cos(x3^(1/4)) + 1.0*sin(x2) + 0.8*x1^2 - 9.98*x3^(1/8) + 113.0

with fit error: 0.0031025

artifical_plus2.csv
mfun =

1.91*x1 - 1.91*cos(x4) - 3.25*exp(x2) + 0.496*exp(x4) + 0.139*x3*exp(x1) + 0.496*x1^(1/2) - 0.496*x2^2 + 1.91*x4^(1/2) + 5.55

with fit error: 0.4703

artifical_times.csv

mfun =

2.56*x1 + 0.854*x2 + 0.854*x4 + 0.0907*x1^2*x4^2 - 0.107*x1*exp(x2) - 0.107*x1*x3*x4^2 + 0.0907*x1^2*x2*x4^2 - 0.0109*x1^2*x2^3*x4^2 + 96.0

with fit error: 0.65413

Cd_wedge.csv

mfun =

3.13e-6*x1*x5*(3.46*x4)^(1/2)*(x3 - 7.14) - 4.35e-8*x1*x4*x5*(x3 - 7.09) - 2.1e-4*x1*x5*(x3 - 7.21) + 0.0267*x1*x2*(x3 - 6.83)*(x3 - 1.0*exp(x1)) + 0.00107

with fit error: 0.0093015

Heatflux_flatplate.csv

mfun =

0.0711*x1 + 1.04e-6*x2 - 2.85e-4*(x1*x2)^(1/2) - 1.16e-4*x2^(1/2) + 0.00514

with fit error: 0.00041841

Heatflux_sphere.csv

mfun =

0.138*(x1*x3 + x1*x5)^(1/2) - 380.0*x1*x2^(1/2)*(x1*x3)^(1/2) + 2.57*x1^2*x2^(1/2)*cos(x2) + 1.77e4*x1*x2^(1/2)*x3*cos(x3) - 3844.0

with fit error: 1354.7769

GSGP

(注意 GSGP 的文本文件需要多两行。
第一行：自变量个数（不含应变量）;
第二行：样本点个数)

C:\Users\rochant\Desktop\SymbolicRegressionAlgorithms\GSGP>a -train_file artifical_plus.txt -test_file artifical_plus.txt
artifical_plus.txt

17.6202
((x1 - x2) * (x2 + x1))

artifical_plus2.txt

3.989
((((((x0 * x2) + (x3 * x1)) / ((x3 + x2) - (x1 + x1))) - (((x0 / x3) - (x0 * x2)) - ((x3 / x1) / (x3 * x0)))) + ((((x2 + x0) - (x1 * x0)) / ((x0 + x0) * (x3 - x0))) + (((x3 + x2) / (x1 * x2)) * ((x0 + x3) + (x0 * x3))))) - (((((x2 - x1) + (x3 + x2)) * ((x3 + x2) - (x0 - x2))) / (((x0 + x2) / (x2 + x0)) * ((x1 - x1) - (x3 * x1)))) * ((((x2 * x2) / (x2 - x2)) * ((x0 + x0) * (x0 / x1))) + (((x3 + x1) * (x0 - x0)) + ((x1 / x0) * (x0 - x2))))))

artifical_times.txt
42.1349
(((((x2 - x1) + (x1 - x2)) + ((x0 * x2) + (x0 / x0))) * (((x0 - x2) - (x0 + x0)) - ((x1 * x0) / (x1 - x1)))) / ((((x0 - x3) + (x1 * x0)) + ((x0 * x3) * (x3 + x1))) * (((x3 * x3) / (x1 / x0)) - ((x2 / x3) * (x0 - x2)))))

Cd_wedge.txt a -train_file Cd_wedge.txt -test_file Cd_wedge.txt

0 (should be NaN, actually)
((((((x1 * x1) - (x2 / x0)) + ((x0 / x4) + (x2 * x3))) / (((x4 - x4) + (x1 + x3)) * ((x3 + x0) - (x0 * x4)))) / ((((x4 * x3) + (x4 + x3)) - ((x4 - x0) / (x0 + x0))) * (((x2 / x0) - (x2 / x0)) * ((x2 + x4) * (x0 / x3)))))) + (((((x3 / x4) - (x0 - x4)) / ((x3 + x4) + (x4 + x4))) * (((x4 * x3) / (x3 * x2)) * ((x4 * x4) / (x2 * x0)))) * ((((x4 + x3) / (x3 + x3)) / ((x1 - x3) - (x4 * x4))) - (((x3 - x1) / (x4 - x4)) * ((x1 + x4) - (x2 * x2)))))))

Heatflux_flatplate.txt

0.000803012
((x1 / x0) - (x1 + x0))

Heatflux_sphere.txt a -train_file Heatflux_sphere.txt -test_file Heatflux_sphere.txt

6594.47
((((((x1 + x0) - (x1 + x3)) - ((x4 - x2) * (x0 / x3))) - (((x4 + x4) - (x3 * x3)) / ((x0 + x0) + (x3 / x4)))) / ((((x2 + x2) * (x3 / x0)) - ((x2 + x4) - (x2 * x0))) + (((x3 * x1) / (x1 + x2)) * ((x1 - x2) - (x3 / x2))))) / (((((x2 + x0) / (x1 * x3)) + ((x0 / x2) / (x3 - x0))) - (((x0 / x0) * (x4 + x2)) + ((x4 - x1) / (x4 / x4)))) * ((((x0 * x2) * (x3 / x2)) * ((x2 + x0) * (x3 * x3))) + (((x4 - x2) / (x2 + x4)) + ((x2 + x3) + (x1 + x0)))))))

SR

artifical_plus.csv
mostAccurate:
(0.1999965861291332 .* (+ (+ (* X1 X1) (+ (mydivide (cos (cos X3)) X3) X4)) (+ (+ (* X1 X1) (+ (sin X2) X4)) (+ (+ (* X1 X1) (+ (+ (* X1 X1) (+ (sin X2) (+ (sin X2) X4))) (+ (sin X2) (+ (sin X2) X4)))) (+ (exp (cos X3)) X4))))) + 100.4787755473883400

artifical_plus2.csv
mostAccurate:
(3.1319943477638796 .* (- (- (- (- (* X4 (exp (cos (sin (exp (cos (sin (sin X1)))))))) (exp X2)) (cos X1)) (sin (sin (sin X3)))) X2)) + 11.5291274776289950

artifical_times.csv

mostAccurate:

(0.3886738851582918 .* (* (+ X1 (- X1 (mydivide X1 X3))) (* (sin X2) (* (+ X1 (+ X1 (- (mydivide X1 (+ (+ (+ X1 (- X1 (mydivide X1 X3))) (- X1 (+ X1 X3))) X3)) (sin X3)))) (mydivide X4 X3))))) + 100.0014980527669500

Cd_wedge.csv

mostAccurate:

(2.9806816126984310 .* (exp (mydivide (* (mydivide (mydivide X1 X3) X3) (- X5 X4)) X4))) + -2.9801206653481094

Heatflux_flatplate.csv

mostAccurate:

(204.6825071242383600 .* (mydivide (* X1 (cos (+ (* X1 (cos (+ (mydivide X1 (cos X2)) X1))) (cos (exp (+ (sin X2) (cos (* (cos (+ (sin X2) (cos X1))) (cos X2)))))))))) (+ (* (exp (+ (exp (cos X1)) (exp (cos (+ (sin X2) (cos (cos (+ (sin X2) (cos X1))))))))) (exp (cos X1))) (+ (exp (exp (exp (cos (exp X1))))) (+ (exp (exp (exp (cos (exp X1))))) X2))))) + 0.0001601501003997

Heatflux_sphere.csv

mostAccurate:

(0.0000011019011184 .* (mydivide (* (* (* (* X2 (- X1 (* X2 (mydivide (* (exp (* X2 X1)) X1) X3)))) (- X1 (* X2 (mydivide (* (exp (* X2 X1)) X1) X3)))) X1) (- X1 (* X2 (mydivide (* (exp (* X2 X1)) X1) X3)))) (sin X3))) + 3242.5354888817938000

MRGP

artifical_plus.csv

mostAccurate:

(+ (+ (quart (sin (exp (* (quart (quart (sin X3))) (* X4 (mydivide X3 X4)))))) (* (quart (- (sin (sin (sin X3))) (quart (- (sin (sin X3)) (sin X2))))) (mydivide X3 (* (sin X2) X3)))) (* (* X3 (* (square X1) (mydivide X4 (square X1)))) (+ (* (mydivide X2 X4) (* (mydivide X3 X4) (+ (square X1) (mydivide X2 X4)))) (quart (sin (exp (* (quart (quart (sin X3))) X3))))))

artifical_plus2.csv

mostAccurate:

(* X4 (+ (* (* (mydivide (cube X1) (quart X2)) (+ (* (sqrt (+ (cos (- (* (square (- X1 X3)) (mylog (- (cos X1) (cube X1)))) X4)) (sin (mydivide (sin (* (cube X3) (sin (- (mydivide (sin X1) X3) (* (cube X2) (mydivide X1 (mydivide (cube (sqrt X1)) X4)))))))) (* X4 X3))))) (exp (cube (sin (sin X4))))) X4)) (exp (cube (sin (sin X4))))) X4))

artifical_times.csv

mostAccurate:

(cos (- (- (mylog (square (mylog (square X1)))) (* (sin (mydivide (- (exp (- X2 X1)) (* (cube X2) X2)) (- (cube (exp X1)) X1))) (* (sqrt (mylog (cube (sin (mydivide (cube (cube (sin (exp (quart (cos (mydivide X4 (+ (mydivide (mylog X4) X1) (- (- (mylog (square (square X3))) (* (sin (mydivide X2 (- (cube (exp X1)) X4))) (* (sqrt (mylog (cube (sin (mydivide (cube (cube (sin (exp (quart (cos (mydivide X4 (+ (mydivide X1 X1) (exp X4))))))))) (mylog X3)))))) (mydivide (* X1 X2) (cube (mylog (* (cos X1) (square (sin (quart (sin X2)))))))))) (cos (cos (quart (quart (cos (* (+ X1 X1) (sqrt (* (cos X3) (* X4 (square (square (+ (mylog (square X1)) (sin X2))))))))))))))))))))) (mylog (mydivide (* X4 X1) (* (- (- (mylog (square (square X3))) (* (mydivide (* X1 X4) (* (cos X3) (square (sin (quart (sin X2)))))) (* (sqrt (mylog (cube (sin X1)))) (mydivide (* X1 X1) (cube (mylog (* (cos X1) (square (sin (quart (sin X2))))))))))) (cos (cos (quart (* (cos X3) (* X4 (square (square (+ (mylog X1) (sin (sin X2)))))))))) (sqrt (* (cos X3) (* X4 (square (square (+ (mylog X1) X2))))))))))))) (mydivide (* X1 X2) (cube (mylog (* (cos X1) (square (sin (quart (sin X2))))))))))) (cos (cos (quart (quart (cos (* (+ X1 X1) (sqrt (* (cos X3) (* X4 (square (square (+ (mylog X1) (sin X2))))))))))))))))

Cd_wedge.csv

mostAccurate

(* (* (* (mydivide X1 X4) X5) (+ X3 (mylog (- (cube (* (mydivide X5 X2) (mydivide (mydivide X1 X3) X4))) (mydivide (* (* (* (* (mydivide X1 X4) X5) (+ (mydivide X1 X4) (mylog (- (cube X3) (mylog X5))))) X5) X3) X4))))) (* (mydivide (* (mydivide X5 X2) (mydivide (mydivide X1 (mylog (- (cube X3) (mylog (* (* (mydivide (mylog (* (mydivide (mydivide X1 (mylog (- (cube X3) (mylog (* (mylog (- (cube X3) (mylog (* (* (mydivide X1 X4) X5) (cube X3))))) X4))))) X4) X5)) X4) X5) X5))))) X4)) X4) X5))

Knee: (* (mydivide X5 X2) (mydivide (mydivide (mydivide X1 X3) X3) X4))

Heatflux_flatplate.csv

mostAccurate:

(mydivide (sqrt (mydivide (sin (sin X1)) (sqrt X2))) (sqrt (mydivide (mylog (mydivide X1 X2)) (sqrt (mydivide (sqrt (mydivide (sin X1) (sqrt (mydivide (mylog X2) (sqrt (mylog (mylog (mylog (mylog X1))))))))) (sqrt (mydivide (mylog (mydivide (sin (mylog X1)) (sqrt (mydivide (mylog (mydivide X1 X2)) (sqrt (mydivide (sqrt (mydivide (sin (sin X1)) (mylog X2))) (sqrt (mydivide (mylog (mylog X2)) (sqrt (mydivide (mydivide X1 X2) X2)))))))))) (sqrt (mydivide (mydivide X1 X2) X2)))))))))

Knee: (sqrt (mydivide X1 (sqrt X2)))

Heatflux_sphere.csv

mostAccurate:

(mydivide (* (cube X1) (mylog (mydivide (mydivide (mydivide (* (mydivide (* (square (* X2 X1)) (mydivide (* X3 (mydivide (* (cube X1) (mydivide X4 X3)) X3)) (- X4 X5))) (mylog (mydivide (mydivide X1 (mylog (quart

(mylog (mydivide (mydivide (mydivide (* (quart (mylog (mydivide (mydivide (mydivide X3 (sin X2)) X1) (sqrt X1)))) (mydivide (* (cube X1) X2) X3)) (sin X2)) X1) (sqrt X1)))))) (sqrt (mydivide (mydivide (* X1 (mydivide (* X2 (cube X1)) X3)) (- X4 X5)) (- X4 X2))))))) (mydivide (* (cube X1) X2) X3)) (sin X2)) X1) (sqrt X1)))) (- (mydivide (mydivide X1 (mylog (mydivide X3 (sin X2)))) (sqrt (mydivide (mydivide (* (cube X1) (mydivide (* X2 X1) X3)) (- X4 X5)) X3))) X5))

Knee: mydivide (* X4 X1) (mylog (mydivide (mydivide X1 (mylog (mydivide X3 (sin X2)))) (sqrt (mydivide (mydivide (* (cube X1) (mydivide (* X2 X1) X3)) (- X4 X5)) (mydivide (* (cube X1) (mydivide (mydivide (mydivide (* X2 (cube X1)) (cube X1)) X5) X3)) X5))))))