COSC 4364 Matlab Problems

Austin Moreau

May 5, 2018

Assignment 1 Problem 9

The code for this problem can be found in the matlab files Assignment1_9.m and Assignment1_9single.m. The output, documented here can be created by opening the files and hitting the run button.

Assignment1_9 single precision output

k = 1, f = 3.18767, |f - pi| = 0.04608, g = 3.14058, |g - pi| = 0.00101319
k = 2, f = 3.14168, |f - pi| = 8.78255e-05, g = 3.14159, |g - pi| = 8.74228e-08
k = 3, f = 3.14159, |f - pi| = 8.74228e-08, g = 3.14159, |g - pi| = 8.74228e-08
k = 4, f = 3.14159, |f - pi| = 8.74228e-08, g = 3.14159, |g - pi| = 8.74228e-08
k = 5, f = 3.14159, |f - pi| = 8.74228e-08, g = 3.14159, |g - pi| = 8.74228e-08

Assignment1_9 double precision output

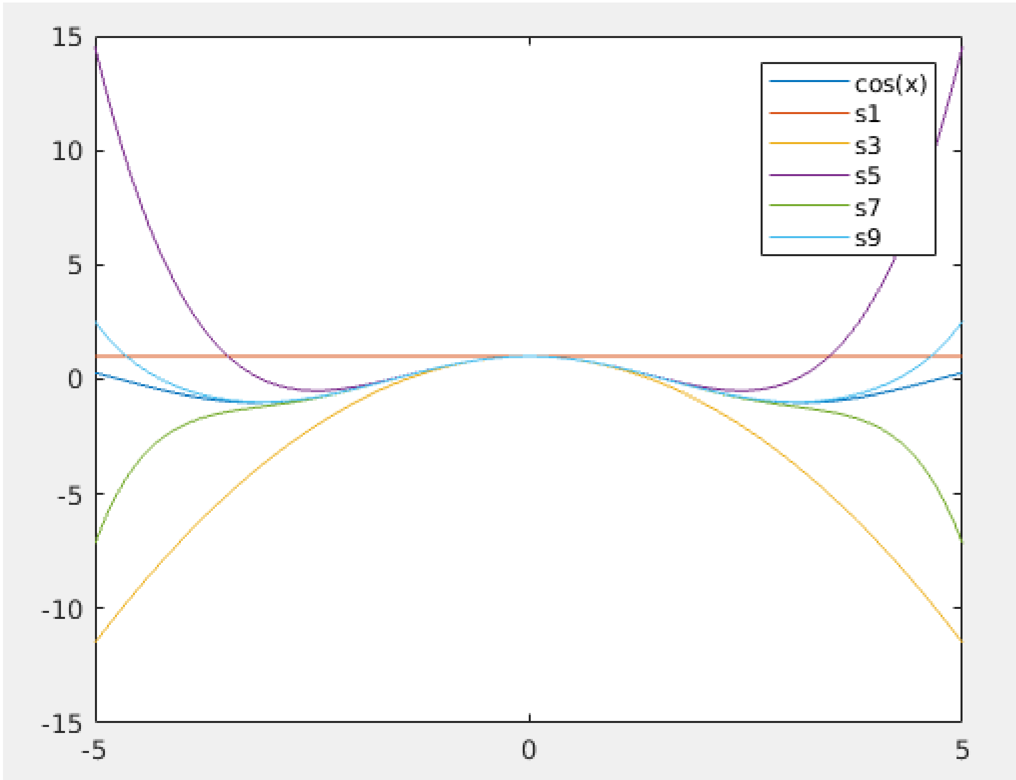
k = 1, f = 3.18767, |f - pi| = 0.04608, g = 3.14058, |g - pi| = 0.0010134
k = 2, f = 3.14168, |f - pi| = 8.76397e-05, g = 3.14159, |g - pi| = 7.37625e-09
k = 3, f = 3.14159, |f - pi| = 3.05654e-10, g = 3.14159, |g - pi| = 8.88178e-16
k = 4, f = 3.14159, |f - pi| = 8.88178e-16, g = 3.14159, |g - pi| = 8.88178e-16
k = 5, f = 3.14159, |f - pi| = 8.88178e-16, g = 3.14159, |g - pi| = 8.88178e-16

Though the output from both problems are incredibly similar, the double precision is just that doubly precise. The number of decimal places required is double that of the single precision output even though both seem to be able to calculate pi in the same number of steps. The differences between the calculated pi and actual pi are again precise to the level of precision required for each problem.

Assignment1 Problem 10

The code for this problem can be found in the matlab file Assignment1_10.m and the output can be recreated by hitting the run button after opening the file. The graph created from running the program is documented below.

As can be expected, higher degrees of Taylor polynomials create graphical representations that are steadily more and more similar to the initial cosine funciton.



Assignment 2 Problem 1

The code for this problem can be found in the matlab file Assignment2_1.m. The solution can be reproduced by opening the file and typing Assignment2_1() into the command line

a.

I. Elapsed time is 3.736409 seconds.
Without Pivoting, single precision
MSE = 0.358544
SMSE = 0.598786

II. Elapsed time is 4.764135 seconds.
Without Pivoting, double precision
MSE = 4.13439e-19
SMSE = 6.42992e-10

b. Elapsed time is 4.905961 seconds.
With Pivoting
MSE = 86.2797
SMSE = 9.28869

c. Using built in funcitons
Elapsed time is 0.036667 seconds.
MSE = 1.01539e-23

SMSE = 3.18652e-12

As could be expected the single precision solver was the quickest of the hand written solvers but the built in function linsolve was by far the quickest solution. Besides being effected by the level of precision the mean squared errors of the problems seem to vary widely.

Assignment 2 Problem 2

The code for this problem is contained in the matlab file Assignment2_2.m. To produce the results simply open the matlab file and type Assignment2_2() into the command line to run the program.

a.

I. Elapsed time is 3.723121 seconds.
With Pivoting, single precision
MSE = 6.39605e+17
SMSE = 7.99753e+08

II. Elapsed time is 4.915473 seconds.
With Pivoting, double precision
MSE = 1.61194e+35
SMSE = 4.0149e+17

b.

Using built in funcitons
Elapsed time is 0.052264 seconds.
MSE = 9.37956e+07
SMSE = 9684.81

Similar to the previous problem, the single precision was the quickest of the hand written solvers but again the built in function linsolve was magnitudes faster. Again in similar fashion to the previous problem, the answers found vary widely.

Assignment 3 Problem 5

a. The solution to this problem can be found in the matlab file labeled Assiggnment3_5bisection.m. To reproduce the results type Assignment3_5bisection(-1,1,100,.5*10^-14) into the command line.

n = 0, c = 0, f(c) = 3, error = 1
n = 1, c = -0.5, f(c) = -2.5, error = 0.5
n = 2, c = -0.25, f(c) = -0.09375, error = 0.25
n = 3, c = -0.125, f(c) = 1.35547, error = 0.125
n = 4, c = -0.1875, f(c) = 0.60791, error = 0.0625
n = 5, c = -0.21875, f(c) = 0.251526, error = 0.03125
n = 6, c = -0.234375, f(c) = 0.0775223, error = 0.015625

n = 7, c = -0.242188, f(c) = -0.00845242, error = 0.0078125
n = 8, c = -0.238281, f(c) = 0.0344499, error = 0.00390625
n = 9, c = -0.240234, f(c) = 0.0129776, error = 0.00195312
n = 10, c = -0.241211, f(c) = 0.00225727, error = 0.000976562
n = 11, c = -0.241699, f(c) = -0.00309889, error = 0.000488281
n = 12, c = -0.241455, f(c) = -0.00042114, error = 0.000244141
n = 13, c = -0.241333, f(c) = 0.000917984, error = 0.00012207
n = 14, c = -0.241394, f(c) = 0.000248401, error = 6.10352e-05
n = 15, c = -0.241425, f(c) = -8.63748e-05, error = 3.05176e-05
n = 16, c = -0.241409, f(c) = 8.10119e-05, error = 1.52588e-05
n = 17, c = -0.241417, f(c) = -2.68178e-06, error = 7.62939e-06
n = 18, c = -0.241413, f(c) = 3.9165e-05, error = 3.8147e-06
n = 19, c = -0.241415, f(c) = 1.82416e-05, error = 1.90735e-06
n = 20, c = -0.241416, f(c) = 7.7799e-06, error = 9.53674e-07
n = 21, c = -0.241416, f(c) = 2.54906e-06, error = 4.76837e-07
n = 22, c = -0.241417, f(c) = -6.63609e-08, error = 2.38419e-07
n = 23, c = -0.241417, f(c) = 1.24135e-06, error = 1.19209e-07
n = 24, c = -0.241417, f(c) = 5.87494e-07, error = 5.96046e-08
n = 25, c = -0.241417, f(c) = 2.60566e-07, error = 2.98023e-08
n = 26, c = -0.241417, f(c) = 9.71027e-08, error = 1.49012e-08
n = 27, c = -0.241417, f(c) = 1.53709e-08, error = 7.45058e-09
n = 28, c = -0.241417, f(c) = -2.5495e-08, error = 3.72529e-09
n = 29, c = -0.241417, f(c) = -5.06208e-09, error = 1.86265e-09
n = 30, c = -0.241417, f(c) = 5.15439e-09, error = 9.31323e-10
n = 31, c = -0.241417, f(c) = 4.61564e-11, error = 4.65661e-10
n = 32, c = -0.241417, f(c) = -2.50796e-09, error = 2.32831e-10
n = 33, c = -0.241417, f(c) = -1.2309e-09, error = 1.16415e-10
n = 34, c = -0.241417, f(c) = -5.92373e-10, error = 5.82077e-11
n = 35, c = -0.241417, f(c) = -2.73109e-10, error = 2.91038e-11
n = 36, c = -0.241417, f(c) = -1.13476e-10, error = 1.45519e-11
n = 37, c = -0.241417, f(c) = -3.36597e-11, error = 7.27596e-12
n = 38, c = -0.241417, f(c) = 6.24834e-12, error = 3.63798e-12
n = 39, c = -0.241417, f(c) = -1.37059e-11, error = 1.81899e-12
n = 40, c = -0.241417, f(c) = -3.72857e-12, error = 9.09495e-13
n = 41, c = -0.241417, f(c) = 1.25988e-12, error = 4.54747e-13
n = 42, c = -0.241417, f(c) = -1.23457e-12, error = 2.27374e-13
n = 43, c = -0.241417, f(c) = 1.28786e-14, error = 1.13687e-13
n = 44, c = -0.241417, f(c) = -6.11067e-13, error = 5.68434e-14
n = 45, c = -0.241417, f(c) = -2.99316e-13, error = 2.84217e-14
n = 46, c = -0.241417, f(c) = -1.43441e-13, error = 1.42109e-14
n = 47, c = -0.241417, f(c) = -6.52811e-14, error = 7.10543e-15
n = 48, c = -0.241417, f(c) = -2.62013e-14, error = 3.55271e-15
convergence>>

b. The code to run the following problem can be found in the file labeled Assignment3_5RegulaFalsi.m. To run the code simply open the file in matlab and hit the run button.

n = 1, c = -0.625, f(c) = -3.50391, error = 2

n = 2, c = -0.432014, f(c) = -1.903, error = 1.625
n = 3, c = -0.33435, f(c) = -0.973121, error = 1.43201
n = 4, c = -0.28621, f(c) = -0.480415, error = 1.33435
n = 5, c = -0.262875, f(c) = -0.232858, error = 1.28621
n = 6, c = -0.251665, f(c) = -0.111841, error = 1.26287
n = 7, c = -0.246304, f(c) = -0.0534795, error = 1.25166
n = 8, c = -0.243746, f(c) = -0.025518, error = 1.2463
n = 9, c = -0.242526, f(c) = -0.0121636, error = 1.24375
n = 10, c = -0.241945, f(c) = -0.00579517, error = 1.24253
n = 11, c = -0.241668, f(c) = -0.00276038, error = 1.24195
n = 12, c = -0.241537, f(c) = -0.00131469, error = 1.24167
n = 13, c = -0.241474, f(c) = -0.00062612, error = 1.24154
n = 14, c = -0.241444, f(c) = -0.00029818, error = 1.24147
n = 15, c = -0.24143, f(c) = -0.000142002, error = 1.24144
n = 16, c = -0.241423, f(c) = -6.76253e-05, error = 1.24143
n = 17, c = -0.24142, f(c) = -3.22049e-05, error = 1.24142
n = 18, c = -0.241418, f(c) = -1.53368e-05, error = 1.24142
n = 19, c = -0.241417, f(c) = -7.30375e-06, error = 1.24142
n = 20, c = -0.241417, f(c) = -3.47823e-06, error = 1.24142
n = 21, c = -0.241417, f(c) = -1.65642e-06, error = 1.24142
n = 22, c = -0.241417, f(c) = -7.88826e-07, error = 1.24142
n = 23, c = -0.241417, f(c) = -3.75658e-07, error = 1.24142
n = 24, c = -0.241417, f(c) = -1.78898e-07, error = 1.24142
n = 25, c = -0.241417, f(c) = -8.51954e-08, error = 1.24142
n = 26, c = -0.241417, f(c) = -4.05721e-08, error = 1.24142
n = 27, c = -0.241417, f(c) = -1.93214e-08, error = 1.24142
n = 28, c = -0.241417, f(c) = -9.20134e-09, error = 1.24142
n = 29, c = -0.241417, f(c) = -4.3819e-09, error = 1.24142
n = 30, c = -0.241417, f(c) = -2.08677e-09, error = 1.24142
n = 31, c = -0.241417, f(c) = -9.9377e-10, error = 1.24142
n = 32, c = -0.241417, f(c) = -4.73258e-10, error = 1.24142
n = 33, c = -0.241417, f(c) = -2.25377e-10, error = 1.24142
n = 34, c = -0.241417, f(c) = -1.0733e-10, error = 1.24142
n = 35, c = -0.241417, f(c) = -5.11133e-11, error = 1.24142
n = 36, c = -0.241417, f(c) = -2.4341e-11, error = 1.24142
n = 37, c = -0.241417, f(c) = -1.15921e-11, error = 1.24142
n = 38, c = -0.241417, f(c) = -5.52047e-12, error = 1.24142
n = 39, c = -0.241417, f(c) = -2.62901e-12, error = 1.24142
n = 40, c = -0.241417, f(c) = -1.25144e-12, error = 1.24142
n = 41, c = -0.241417, f(c) = -5.95968e-13, error = 1.24142
n = 42, c = -0.241417, f(c) = -2.83773e-13, error = 1.24142
n = 43, c = -0.241417, f(c) = -1.35003e-13, error = 1.24142
n = 44, c = -0.241417, f(c) = -6.43929e-14, error = 1.24142
n = 45, c = -0.241417, f(c) = -3.10862e-14, error = 1.24142
n = 46, c = -0.241417, f(c) = -1.5099e-14, error = 1.24142
n = 47, c = -0.241417, f(c) = -6.66134e-15, error = 1.24142
n = 48, c = -0.241417, f(c) = -3.10862e-15, error = 1.24142
n = 49, c = -0.241417, f(c) = -1.33227e-15, error = 1.24142
n = 50, c = -0.241417, f(c) = -8.88178e-16, error = 1.24142

n = 51, c = -0.241417, f(c) = -4.44089e-16, error = 1.24142

c. I did not complete this problem

d. The code to run the following problem can be found in the file labeled Assignment3_5Newton.m. To run the code simply open the file in matlab and hit the run button. The initial starting point for these problems is the x at the top of the tables. The initial starting points do not seem to effect the amount of steps necessary for the root to be found. I would assume this is likely because the interval (-1, 1) is such a small space that to find the root over this interval will not take many steps.

x = 0, f(x) = 3
n = 1, x = -0.214286, f(x) = 3
n = 2, x = -0.241047, f(x) = 0.301749
n = 3, x = -0.241417, f(x) = 0.00405411
n = 4, x = -0.241417, f(x) = 7.5786e-07
n = 5, x = -0.241417, f(x) = 2.66454e-14

x = 5.000000e-01, f(x) = 12
n = 1, x = -0.0333333, f(x) = 12
n = 2, x = -0.221002, f(x) = 2.54104
n = 3, x = -0.241207, f(x) = 0.226274
n = 4, x = -0.241417, f(x) = 0.00229982
n = 5, x = -0.241417, f(x) = 2.43937e-07
n = 6, x = -0.241417, f(x) = 2.66454e-15

x = -8.000000e-01, f(x) = -4.744
n = 1, x = -0.0855422, f(x) = -4.744
n = 2, x = -0.229738, f(x) = 1.85238
n = 3, x = -0.241348, f(x) = 0.128878
n = 4, x = -0.241417, f(x) = 0.000754651
n = 5, x = -0.241417, f(x) = 2.62701e-08
n = 6, x = -0.241417, f(x) = 0

e. The code to run the following problem can be found in the file labeled Assignment3_5secant.m. To run the code simply open the file in matlab and hit the run button. Again, similar to the Newton method, the number of iterations required only vary by one between the three starting points, -1, 0, and 1. Again I'd assume this is because the interval is so small.

Initial starting point x = b
iter: 0 a: 1.0000 fa: 26.0000 b: -1.0000 fb: -6.0000 x: -0.6250 fx: -3.5039 deltaX: 0.3750
iter: 1 a: -1.0000 fa: -6.0000 b: -0.6250 fb: -3.5039 x: -0.0986 fx: 1.6858 deltaX: 0.5264
iter: 2 a: -0.6250 fa: -3.5039 b: -0.0986 fb: 1.6858 x: -0.2696 fx: -0.3047 deltaX: -0.1710
iter: 3 a: -0.0986 fa: 1.6858 b: -0.2696 fb: -0.3047 x: -0.2434 fx: -0.0219 deltaX: 0.0262
iter: 4 a: -0.2696 fa: -0.3047 b: -0.2434 fb: -0.0219 x: -0.2414 fx: 0.0003 deltaX: 0.0020
iter: 5 a: -0.2434 fa: -0.0219 b: -0.2414 fb: 0.0003 x: -0.2414 fx: -0.0000 deltaX: -0.0000
iter: 6 a: -0.2414 fa: 0.0003 b: -0.2414 fb: -0.0000 x: -0.2414 fx: -0.0000 deltaX: 0.0000
iter: 7 a: -0.2414 fa: -0.0000 b: -0.2414 fb: -0.0000 x: -0.2414 fx: 0.0000 deltaX: 0.0000
iter: 8 a: -0.2414 fa: -0.0000 b: -0.2414 fb: 0.0000 x: -0.2414 fx: 0.0000 deltaX: -0.0000

initial starting point x = a
iter: 0 a: 1.0000 fa: 26.0000 b: -1.0000 fb: -6.0000 x: 1.3750 fx: 40.6836 deltaX: 0.3750
iter: 1 a: -1.0000 fa: -6.0000 b: 1.3750 fb: 40.6836 x: -0.6948 fx: -4.0185 deltaX: -2.0698
iter: 2 a: 1.3750 fa: 40.6836 b: -0.6948 fb: -4.0185 x: -0.5087 fx: -2.5736 deltaX: 0.1861
iter: 3 a: -0.6948 fa: -4.0185 b: -0.5087 fb: -2.5736 x: -0.1773 fx: 0.7269 deltaX: 0.3314
iter: 4 a: -0.5087 fa: -2.5736 b: -0.1773 fb: 0.7269 x: -0.2503 fx: -0.0967 deltaX: -0.0730
iter: 5 a: -0.1773 fa: 0.7269 b: -0.2503 fb: -0.0967 x: -0.2417 fx: -0.0031 deltaX: 0.0086
iter: 6 a: -0.2503 fa: -0.0967 b: -0.2417 fb: -0.0031 x: -0.2414 fx: 0.0000 deltaX: 0.0003
iter: 7 a: -0.2417 fa: -0.0031 b: -0.2414 fb: 0.0000 x: -0.2414 fx: -0.0000 deltaX: -0.0000
iter: 8 a: -0.2414 fa: 0.0000 b: -0.2414 fb: -0.0000 x: -0.2414 fx: -0.0000 deltaX: 0.0000
iter: 9 a: -0.2414 fa: -0.0000 b: -0.2414 fb: -0.0000 x: -0.2414 fx: 0.0000 deltaX: 0.0000

initial starting point x = 0
iter: 0 a: 1.0000 fa: 26.0000 b: -1.0000 fb: -6.0000 x: 0.3750 fx: 9.3398 deltaX: 0.3750
iter: 1 a: -1.0000 fa: -6.0000 b: 0.3750 fb: 9.3398 x: -0.4622 fx: -2.1727 deltaX: -0.8372
iter: 2 a: 0.3750 fa: 9.3398 b: -0.4622 fb: -2.1727 x: -0.3042 fx: -0.6672 deltaX: 0.1580
iter: 3 a: -0.4622 fa: -2.1727 b: -0.3042 fb: -0.6672 x: -0.2342 fx: 0.0798 deltaX: 0.0700
iter: 4 a: -0.3042 fa: -0.6672 b: -0.2342 fb: 0.0798 x: -0.2416 fx: -0.0025 deltaX: -0.0075
iter: 5 a: -0.2342 fa: 0.0798 b: -0.2416 fb: -0.0025 x: -0.2414 fx: -0.0000 deltaX: 0.0002
iter: 6 a: -0.2416 fa: -0.0025 b: -0.2414 fb: -0.0000 x: -0.2414 fx: 0.0000 deltaX: 0.0000
iter: 7 a: -0.2414 fa: -0.0000 b: -0.2414 fb: 0.0000 x: -0.2414 fx: -0.0000 deltaX: -0.0000
iter: 8 a: -0.2414 fa: 0.0000 b: -0.2414 fb: -0.0000 x: -0.2414 fx: 0.0000 deltaX: 0.0000

Assignment 3 Problem 6

For this problem I chose a fairly high initial x value to begin with to make any variations in the number of necessary iterations obvious while allowing for a reasonable run time. To reproduce the results documented here simply open the matlab file Assignment3_6Newton.m and hit the run button. As can be expected for a process such as this, the number of steps grow exponentially as the initial value increases. The algorithm used in b is always quicker with larger r's producing faster convergence also. Presumably the number of steps for necessary convergence is directly related to the step size. Of course a larger r value produces a larger difference between iterations and problem b, in the way it is designed (the fastest growing value between the two algorithms is the 3R) will grow far quicker than the algorithm used in a.

Problem 6 1

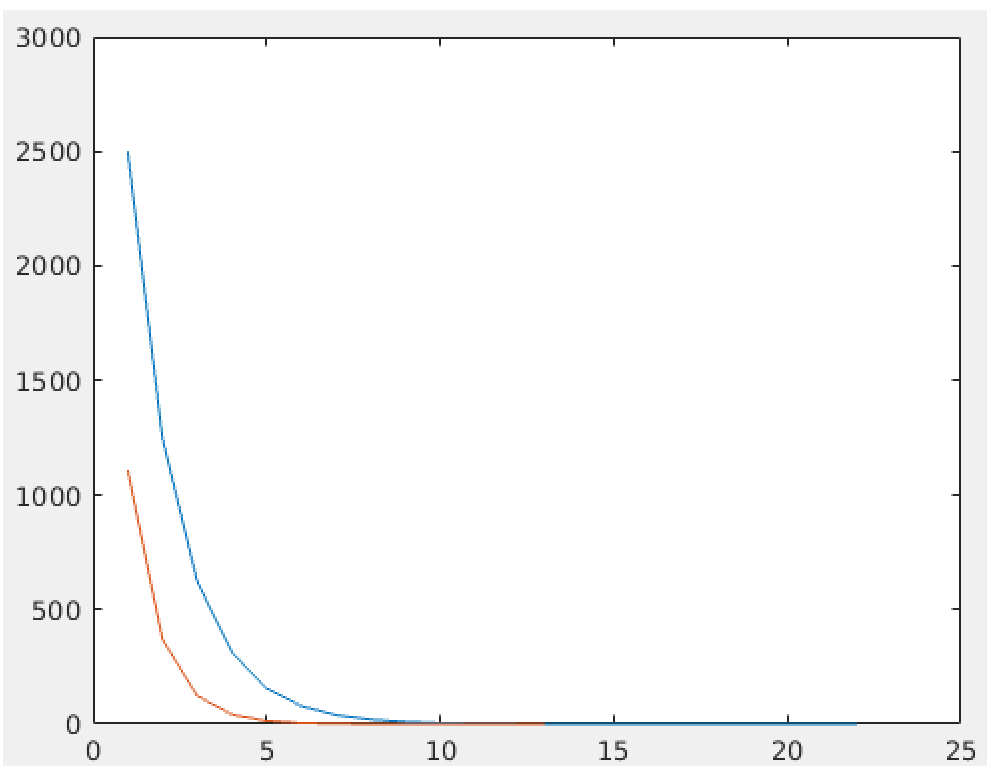
a

iter: 000	x = 10000	fx = 2500	r = 0.001
iter: 001	x = 10000	fx = 1250	r = 0.001
iter: 002	x = 10000	fx = 625	r = 0.001
iter: 003	x = 10000	fx = 312.5	r = 0.001
iter: 004	x = 10000	fx = 156.25	r = 0.001
iter: 005	x = 10000	fx = 78.125	r = 0.001
iter: 006	x = 10000	fx = 39.0625	r = 0.001
iter: 007	x = 10000	fx = 19.5313	r = 0.001
iter: 008	x = 10000	fx = 9.76566	r = 0.001
iter: 009	x = 10000	fx = 4.88288	r = 0.001

iter: 010	x = 10000	fx = 2.44154	r = 0.001
iter: 011	x = 10000	fx = 1.22098	r = 0.001
iter: 012	x = 10000	fx = 0.610898	r = 0.001
iter: 013	x = 10000	fx = 0.306267	r = 0.001
iter: 014	x = 10000	fx = 0.154766	r = 0.001
iter: 015	x = 10000	fx = 0.0806138	r = 0.001
iter: 016	x = 10000	fx = 0.0465093	r = 0.001
iter: 017	x = 10000	fx = 0.0340052	r = 0.001
iter: 018	x = 10000	fx = 0.0317062	r = 0.001
iter: 019	x = 10000	fx = 0.0316229	r = 0.001
iter: 020	x = 10000	fx = 0.0316228	r = 0.001
iter: 021	x = 10000	fx = 0.0316228	r = 0.001

b

iter: 000	x = 10000	fx = 1111.11	r = 0.001
iter: 001	x = 10000	fx = 370.37	r = 0.001
iter: 002	x = 10000	fx = 123.457	r = 0.001
iter: 003	x = 10000	fx = 41.1523	r = 0.001
iter: 004	x = 10000	fx = 13.7174	r = 0.001
iter: 005	x = 10000	fx = 4.57255	r = 0.001
iter: 006	x = 10000	fx = 1.52438	r = 0.001
iter: 007	x = 10000	fx = 0.508709	r = 0.001
iter: 008	x = 10000	fx = 0.171315	r = 0.001
iter: 009	x = 10000	fx = 0.0622352	r = 0.001
iter: 010	x = 10000	fx = 0.033896	r = 0.001
iter: 011	x = 10000	fx = 0.0316254	r = 0.001
iter: 012	x = 10000	fx = 0.0316228	r = 0.001

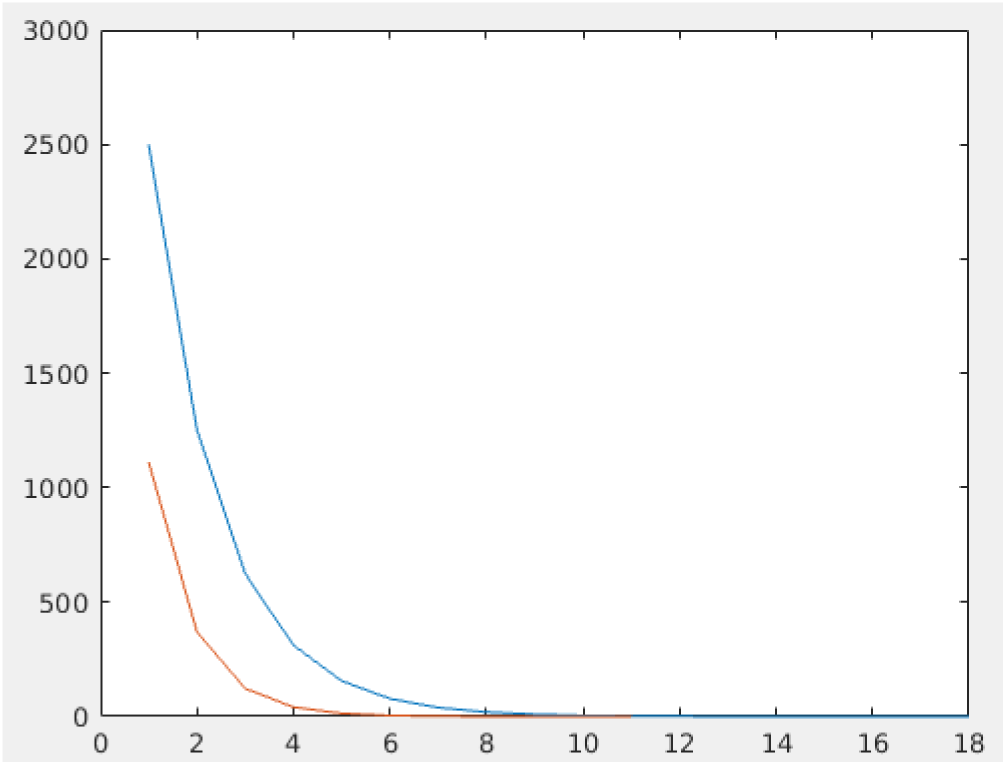


a

iter: 000	x = 10000	fx = 2500	r = 0.1
iter: 001	x = 10000	fx = 1250	r = 0.1
iter: 002	x = 10000	fx = 625	r = 0.1
iter: 003	x = 10000	fx = 312.5	r = 0.1
iter: 004	x = 10000	fx = 156.25	r = 0.1
iter: 005	x = 10000	fx = 78.1254	r = 0.1
iter: 006	x = 10000	fx = 39.0634	r = 0.1
iter: 007	x = 10000	fx = 19.533	r = 0.1
iter: 008	x = 10000	fx = 9.76904	r = 0.1
iter: 009	x = 10000	fx = 4.88964	r = 0.1
iter: 010	x = 10000	fx = 2.45504	r = 0.1
iter: 011	x = 10000	fx = 1.24789	r = 0.1
iter: 012	x = 10000	fx = 0.664012	r = 0.1
iter: 013	x = 10000	fx = 0.407306	r = 0.1
iter: 014	x = 10000	fx = 0.326411	r = 0.1
iter: 015	x = 10000	fx = 0.316387	r = 0.1
iter: 016	x = 10000	fx = 0.316228	r = 0.1
iter: 017	x = 10000	fx = 0.316228	r = 0.1

b

iter: 000	x = 10000	fx = 1111.11	r = 0.1
iter: 001	x = 10000	fx = 370.37	r = 0.1
iter: 002	x = 10000	fx = 123.457	r = 0.1
iter: 003	x = 10000	fx = 41.1531	r = 0.1
iter: 004	x = 10000	fx = 13.7199	r = 0.1
iter: 005	x = 10000	fx = 4.57976	r = 0.1
iter: 006	x = 10000	fx = 1.54597	r = 0.1
iter: 007	x = 10000	fx = 0.572028	r = 0.1
iter: 008	x = 10000	fx = 0.331702	r = 0.1
iter: 009	x = 10000	fx = 0.316236	r = 0.1
iter: 010	x = 10000	fx = 0.316228	r = 0.1

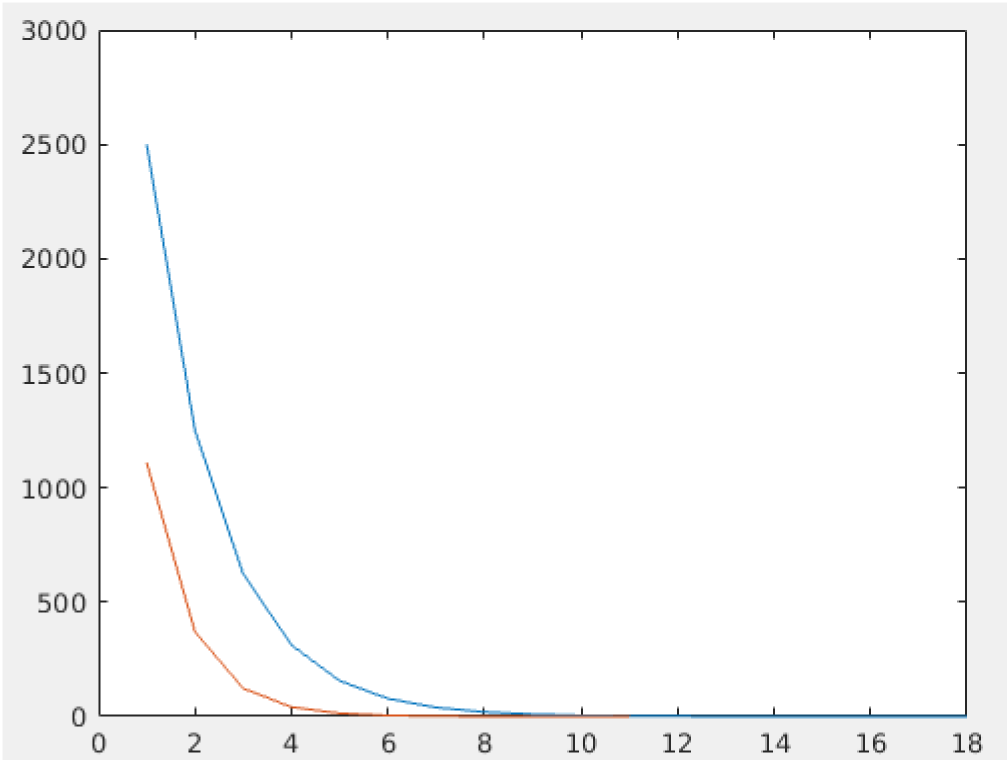


a

iter: 000	x = 10000	fx = 2500	r = 10
iter: 001	x = 10000	fx = 1250	r = 10
iter: 002	x = 10000	fx = 625.005	r = 10
iter: 003	x = 10000	fx = 312.511	r = 10
iter: 004	x = 10000	fx = 156.271	r = 10
iter: 005	x = 10000	fx = 78.1677	r = 10
iter: 006	x = 10000	fx = 39.1478	r = 10
iter: 007	x = 10000	fx = 19.7016	r = 10
iter: 008	x = 10000	fx = 10.1046	r = 10
iter: 009	x = 10000	fx = 5.54712	r = 10
iter: 010	x = 10000	fx = 3.67493	r = 10
iter: 011	x = 10000	fx = 3.19804	r = 10
iter: 012	x = 10000	fx = 3.16248	r = 10
iter: 013	x = 10000	fx = 3.16228	r = 10
iter: 014	x = 10000	fx = 3.16228	r = 10

b

iter: 000	x = 10000	fx = 1111.11	r = 10
iter: 001	x = 10000	fx = 370.379	r = 10
iter: 002	x = 10000	fx = 123.484	r = 10
iter: 003	x = 10000	fx = 41.2332	r = 10
iter: 004	x = 10000	fx = 13.9596	r = 10
iter: 005	x = 10000	fx = 5.27924	r = 10
iter: 006	x = 10000	fx = 3.26362	r = 10
iter: 007	x = 10000	fx = 3.1623	r = 10
iter: 008	x = 10000	fx = 3.16228	r = 10

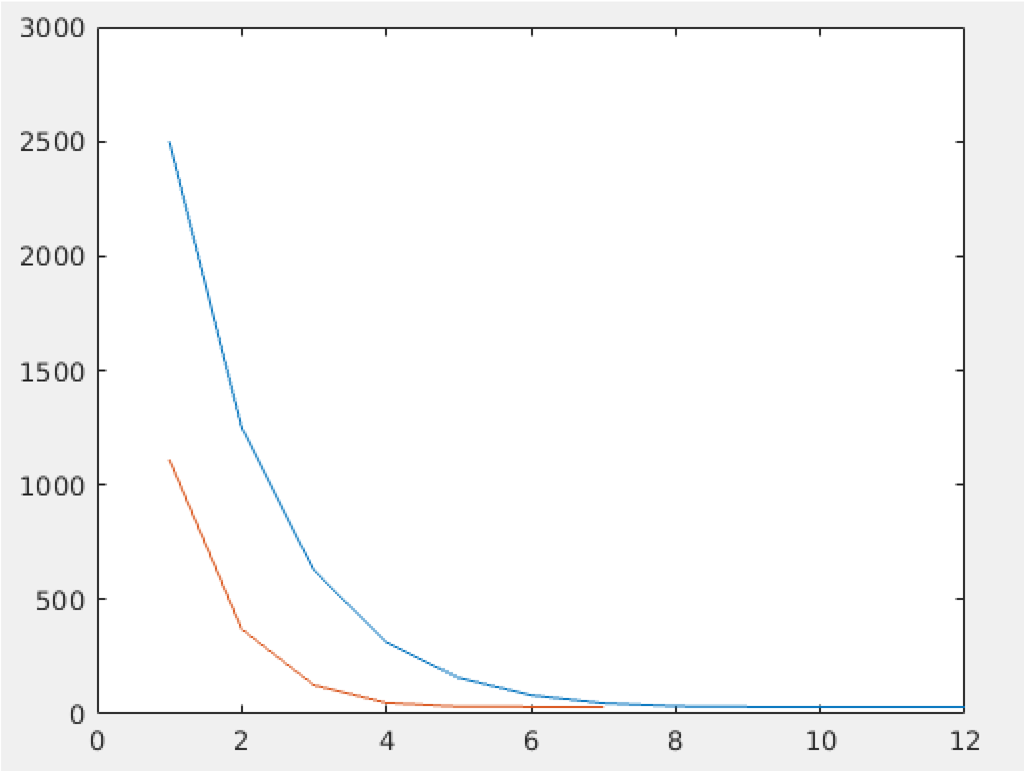


a

iter: 000	x = 10000	fx = 2500.12	r = 1000
iter: 001	x = 10000	fx = 1250.26	r = 1000
iter: 002	x = 10000	fx = 625.531	r = 1000
iter: 003	x = 10000	fx = 313.565	r = 1000
iter: 004	x = 10000	fx = 158.377	r = 1000
iter: 005	x = 10000	fx = 82.3455	r = 1000
iter: 006	x = 10000	fx = 47.2447	r = 1000
iter: 007	x = 10000	fx = 34.2056	r = 1000
iter: 008	x = 10000	fx = 31.7203	r = 1000
iter: 009	x = 10000	fx = 31.6229	r = 1000
iter: 010	x = 10000	fx = 31.6228	r = 1000
iter: 011	x = 10000	fx = 31.6228	r = 1000

b

iter: 000	x = 10000	fx = 1111.41	r = 1000
iter: 001	x = 10000	fx = 371.269	r = 1000
iter: 002	x = 10000	fx = 126.145	r = 1000
iter: 003	x = 10000	fx = 48.9502	r = 1000
iter: 004	x = 10000	fx = 32.2581	r = 1000
iter: 005	x = 10000	fx = 31.6228	r = 1000
iter: 006	x = 10000	fx = 31.6228	r = 1000



Problem 6 2

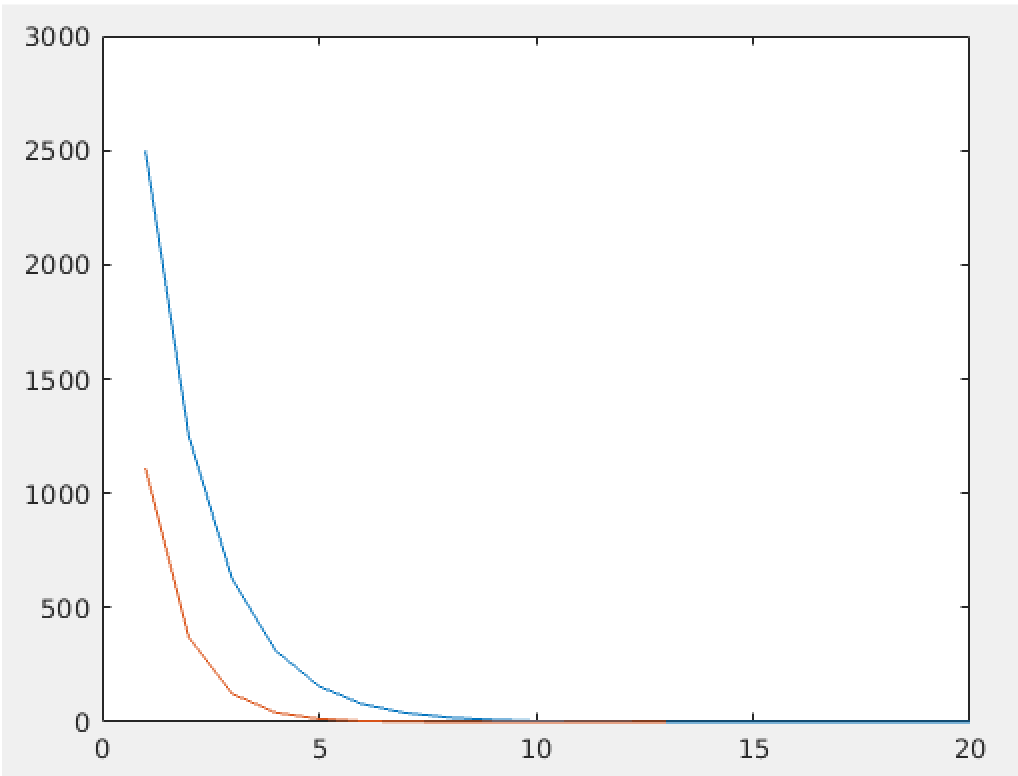
a

iter: 000	x = 10000	fx = 2500	r = 0.001
iter: 001	x = 10000	fx = 1250	r = 0.001
iter: 002	x = 10000	fx = 625	r = 0.001
iter: 003	x = 10000	fx = 312.5	r = 0.001
iter: 004	x = 10000	fx = 156.25	r = 0.001
iter: 005	x = 10000	fx = 78.125	r = 0.001
iter: 006	x = 10000	fx = 39.0625	r = 0.001
iter: 007	x = 10000	fx = 19.5313	r = 0.001
iter: 008	x = 10000	fx = 9.76566	r = 0.001
iter: 009	x = 10000	fx = 4.88288	r = 0.001
iter: 010	x = 10000	fx = 2.44154	r = 0.001
iter: 011	x = 10000	fx = 1.22098	r = 0.001
iter: 012	x = 10000	fx = 0.610898	r = 0.001
iter: 013	x = 10000	fx = 0.306267	r = 0.001
iter: 014	x = 10000	fx = 0.154766	r = 0.001
iter: 015	x = 10000	fx = 0.0806138	r = 0.001
iter: 016	x = 10000	fx = 0.0465093	r = 0.001
iter: 017	x = 10000	fx = 0.0340052	r = 0.001
iter: 018	x = 10000	fx = 0.0317062	r = 0.001
iter: 019	x = 10000	fx = 0.0316229	r = 0.001

b

iter: 000	x = 10000	fx = 1111.11	r = 0.001

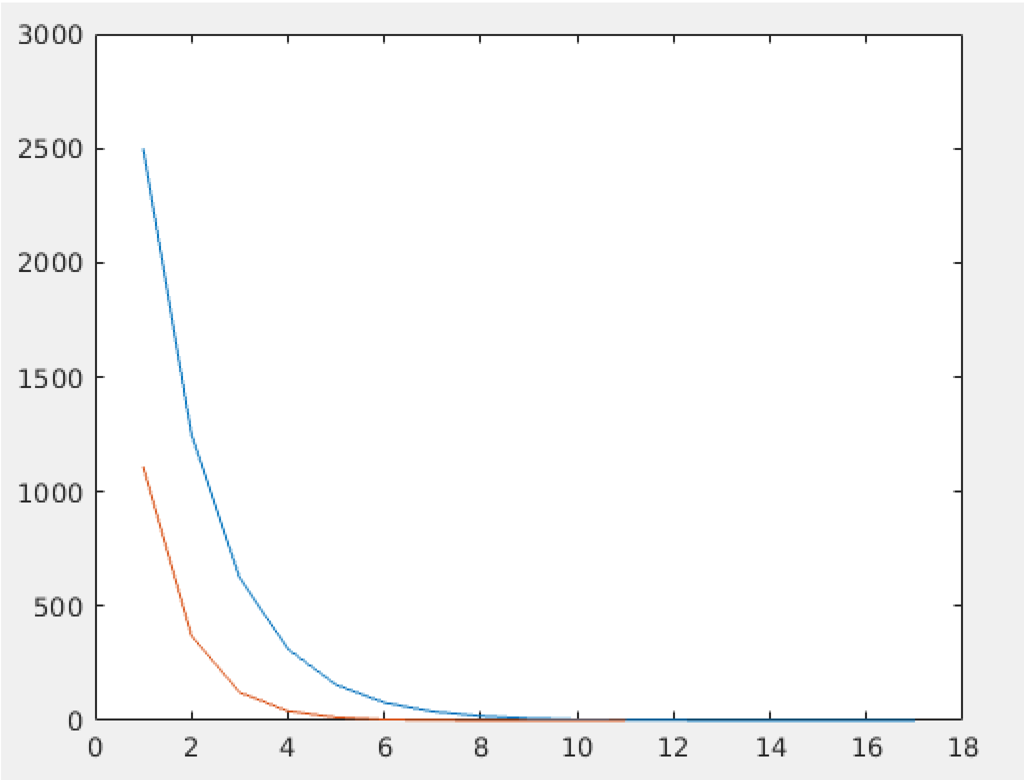
iter: 001	x = 10000	fx = 370.37	r = 0.001
iter: 002	x = 10000	fx = 123.457	r = 0.001
iter: 003	x = 10000	fx = 41.1523	r = 0.001
iter: 004	x = 10000	fx = 13.7174	r = 0.001
iter: 005	x = 10000	fx = 4.57255	r = 0.001
iter: 006	x = 10000	fx = 1.52438	r = 0.001
iter: 007	x = 10000	fx = 0.508709	r = 0.001
iter: 008	x = 10000	fx = 0.171315	r = 0.001
iter: 009	x = 10000	fx = 0.0622352	r = 0.001
iter: 010	x = 10000	fx = 0.033896	r = 0.001
iter: 011	x = 10000	fx = 0.0316254	r = 0.001
iter: 012	x = 10000	fx = 0.0316228	r = 0.001



a

iter: 000	x = 10000	fx = 2500	r = 0.1
iter: 001	x = 10000	fx = 1250	r = 0.1
iter: 002	x = 10000	fx = 625	r = 0.1
iter: 003	x = 10000	fx = 312.5	r = 0.1
iter: 004	x = 10000	fx = 156.25	r = 0.1
iter: 005	x = 10000	fx = 78.1254	r = 0.1
iter: 006	x = 10000	fx = 39.0634	r = 0.1
iter: 007	x = 10000	fx = 19.533	r = 0.1
iter: 008	x = 10000	fx = 9.76904	r = 0.1
iter: 009	x = 10000	fx = 4.88964	r = 0.1
iter: 010	x = 10000	fx = 2.45504	r = 0.1

iter: 011 x = 10000 fx = 1.24789 r = 0.1
iter: 012 x = 10000 fx = 0.664012 r = 0.1
iter: 013 x = 10000 fx = 0.407306 r = 0.1
iter: 014 x = 10000 fx = 0.326411 r = 0.1
iter: 015 x = 10000 fx = 0.316387 r = 0.1
iter: 016 x = 10000 fx = 0.316228 r = 0.1
b
iter: 000 x = 10000 fx = 1111.11 r = 0.1
iter: 001 x = 10000 fx = 370.37 r = 0.1
iter: 002 x = 10000 fx = 123.457 r = 0.1
iter: 003 x = 10000 fx = 41.1531 r = 0.1
iter: 004 x = 10000 fx = 13.7199 r = 0.1
iter: 005 x = 10000 fx = 4.57976 r = 0.1
iter: 006 x = 10000 fx = 1.54597 r = 0.1
iter: 007 x = 10000 fx = 0.572028 r = 0.1
iter: 008 x = 10000 fx = 0.331702 r = 0.1
iter: 009 x = 10000 fx = 0.316236 r = 0.1
iter: 010 x = 10000 fx = 0.316228 r = 0.1

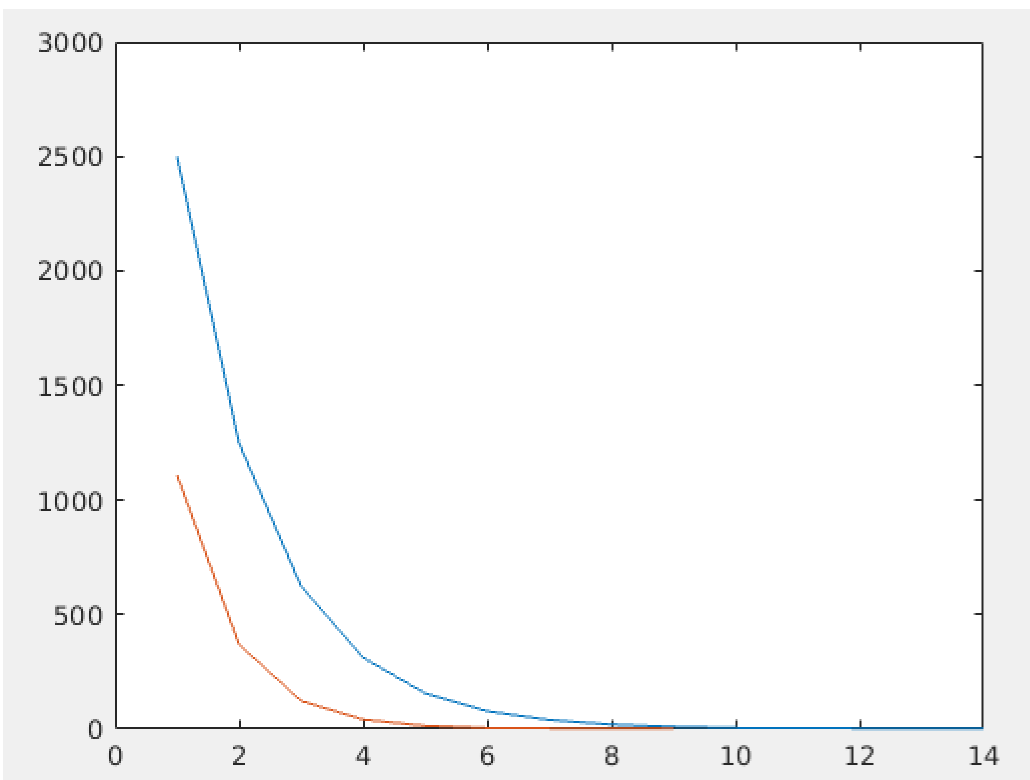


a
iter: 000 x = 10000 fx = 2500 r = 10
iter: 001 x = 10000 fx = 1250 r = 10
iter: 002 x = 10000 fx = 625.005 r = 10
iter: 003 x = 10000 fx = 312.511 r = 10
iter: 004 x = 10000 fx = 156.271 r = 10
iter: 005 x = 10000 fx = 78.1677 r = 10

iter: 006	x = 10000	fx = 39.1478	r = 10
iter: 007	x = 10000	fx = 19.7016	r = 10
iter: 008	x = 10000	fx = 10.1046	r = 10
iter: 009	x = 10000	fx = 5.54712	r = 10
iter: 010	x = 10000	fx = 3.67493	r = 10
iter: 011	x = 10000	fx = 3.19804	r = 10
iter: 012	x = 10000	fx = 3.16248	r = 10
iter: 013	x = 10000	fx = 3.16228	r = 10

b

iter: 000	x = 10000	fx = 1111.11	r = 10
iter: 001	x = 10000	fx = 370.379	r = 10
iter: 002	x = 10000	fx = 123.484	r = 10
iter: 003	x = 10000	fx = 41.2332	r = 10
iter: 004	x = 10000	fx = 13.9596	r = 10
iter: 005	x = 10000	fx = 5.27924	r = 10
iter: 006	x = 10000	fx = 3.26362	r = 10
iter: 007	x = 10000	fx = 3.1623	r = 10
iter: 008	x = 10000	fx = 3.16228	r = 10



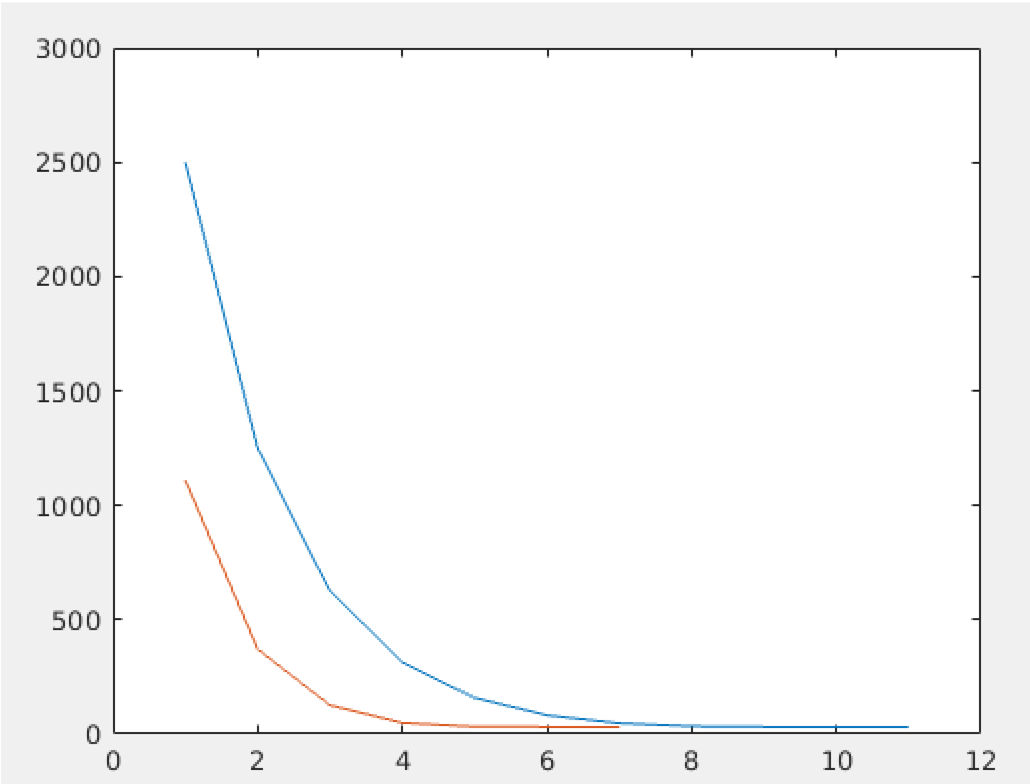
a

iter: 000	x = 10000	fx = 2500.12	r = 1000
iter: 001	x = 10000	fx = 1250.26	r = 1000
iter: 002	x = 10000	fx = 625.531	r = 1000

iter: 003	x = 10000	fx = 313.565	r = 1000
iter: 004	x = 10000	fx = 158.377	r = 1000
iter: 005	x = 10000	fx = 82.3455	r = 1000
iter: 006	x = 10000	fx = 47.2447	r = 1000
iter: 007	x = 10000	fx = 34.2056	r = 1000
iter: 008	x = 10000	fx = 31.7203	r = 1000
iter: 009	x = 10000	fx = 31.6229	r = 1000
iter: 010	x = 10000	fx = 31.6228	r = 1000

b

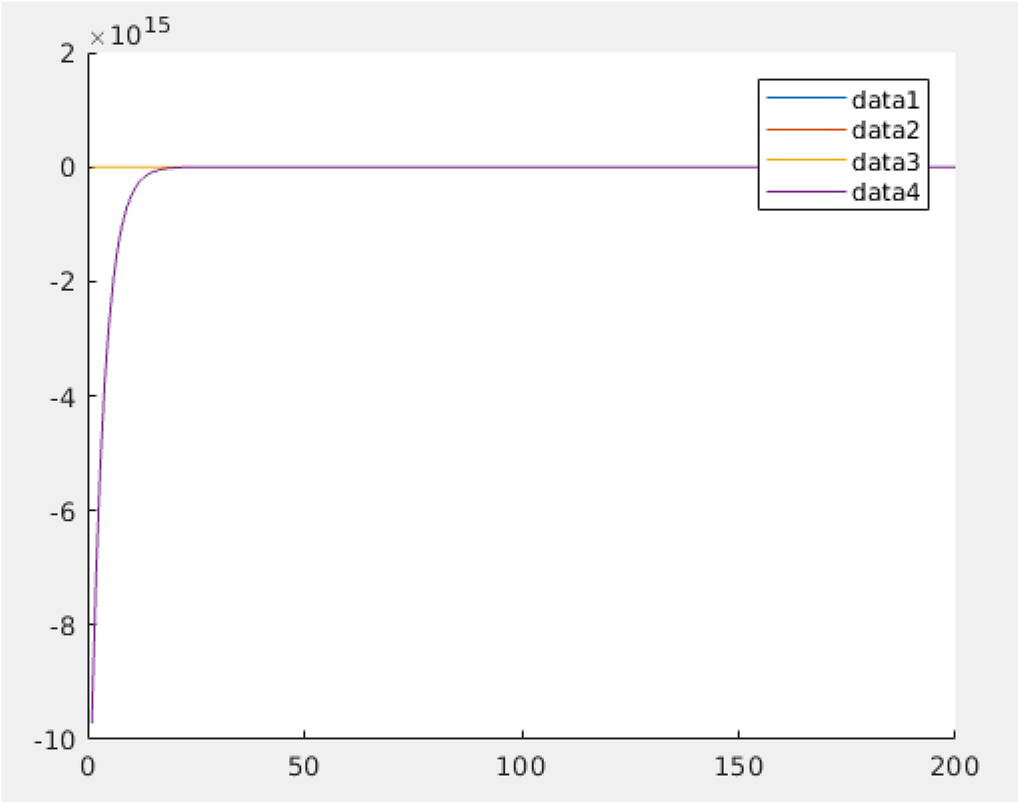
iter: 000	x = 10000	fx = 1111.41	r = 1000
iter: 001	x = 10000	fx = 371.269	r = 1000
iter: 002	x = 10000	fx = 126.145	r = 1000
iter: 003	x = 10000	fx = 48.9502	r = 1000
iter: 004	x = 10000	fx = 32.2581	r = 1000
iter: 005	x = 10000	fx = 31.6228	r = 1000
iter: 006	x = 10000	fx = 31.6228	r = 1000

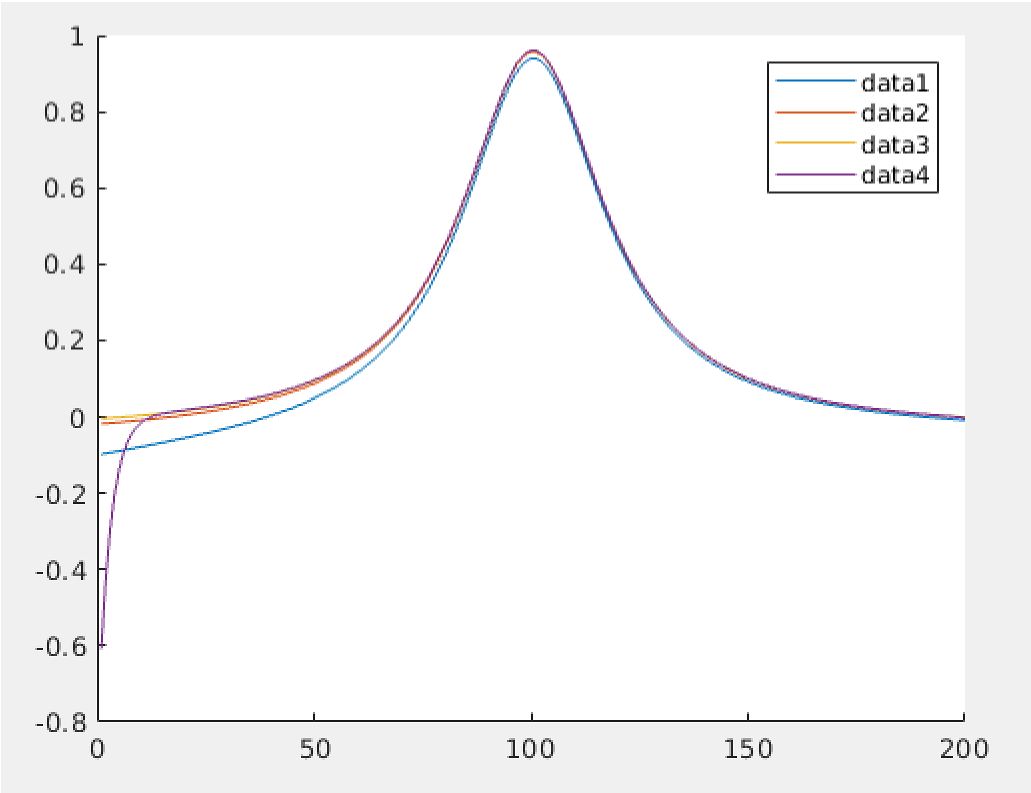


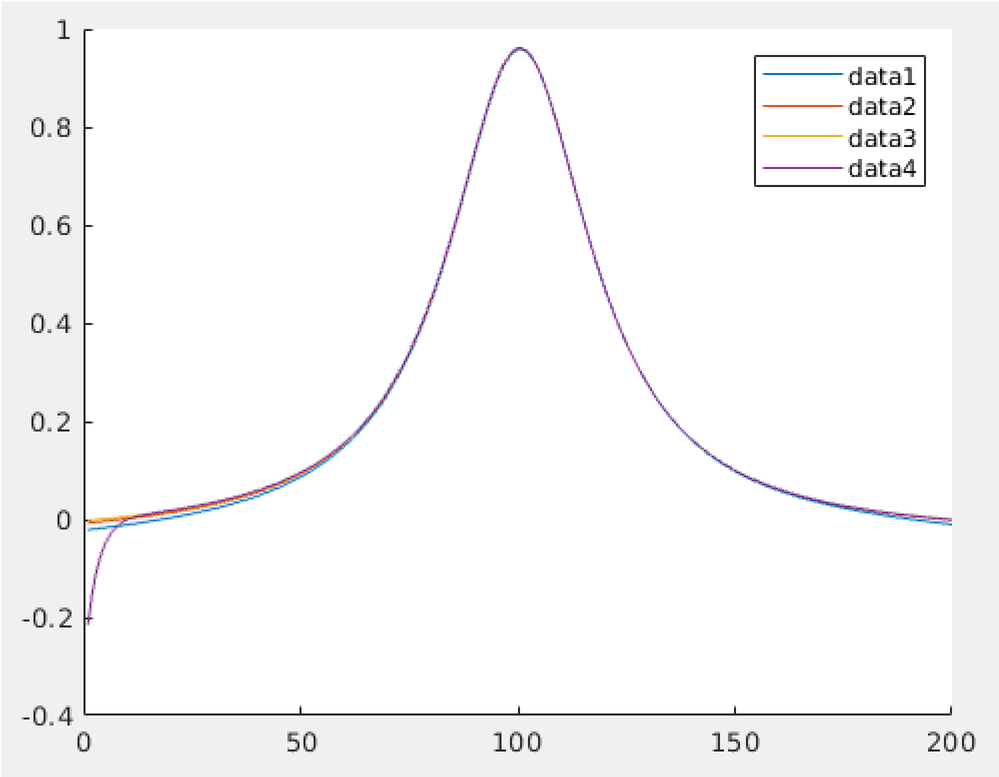
Assignment 4 Problem 8

To reproduce the plots presented here, simply open the matlab file Assignment4_8.m and hit the run button. For the graphs, the the blue line (data 1) represents an n value of 5, the red line (data 2) represent an n value of 10, the yellow line (data 3) an n value of 20 and the purple line (data 4) an n value of 40. The following plots map the error between the actual value and the interpolated value For all three plots there is maximum error value of about 1. For the first plot there a maximum negative error of -10*10^15, this is probably an error. For the second plot, the maximum negative error value is

-.6. For the third plot the maximum negative error value is about -.2. The plots are presented in the order of the following, the first is a, the second is b and the third is c.







Assignment 4 Problem 9

The following results can be reproduced by running the matlab file assignment4_9.m. If a column turns to zeros then the number meant to be there cannot be represented in the format used.

cos

h	forward	centered
0.010000000000000	0.004999958333474	0
0.000100000000000	0.000049999999696	0
0.000001000000000	0.000000500044450	0
0.000000010000000	0	0
0.000000000100000	0	0

arctan

h	forward	centered
0.010000000000000	0.002491666914583	0.000008333083329
0.000100000000000	0.000024999166126	0.000000000833167
0.000001000000000	0.000000250063358	0.000000000041133
0.000000010000000	0.000000003038735	0.000000003038735
0.000000000100000	0.000000041370185	0.000000041370185

Richardson cos

h	forward	centered
0.010000000000000	0.004999958333474	0.004999958333474
0.000100000000000	0.000049999999696	0.000049999999696
0.000001000000000	0.000000500044450	0.000000500044450
0.000000010000000	0	0
0.000000000100000	0	0

Richardson arctan

h	forward	centered
0.010000000000000	0.002491666914583	0.002491666914583
0.000100000000000	0.000024999166126	0.000024999166126
0.000001000000000	0.000000250063358	0.000000250063358
0.000000010000000	0.000000003038735	0.000000003038735
0.000000000100000	0.000000041370185	0.000000041370185

Assignment 6 Problem 6

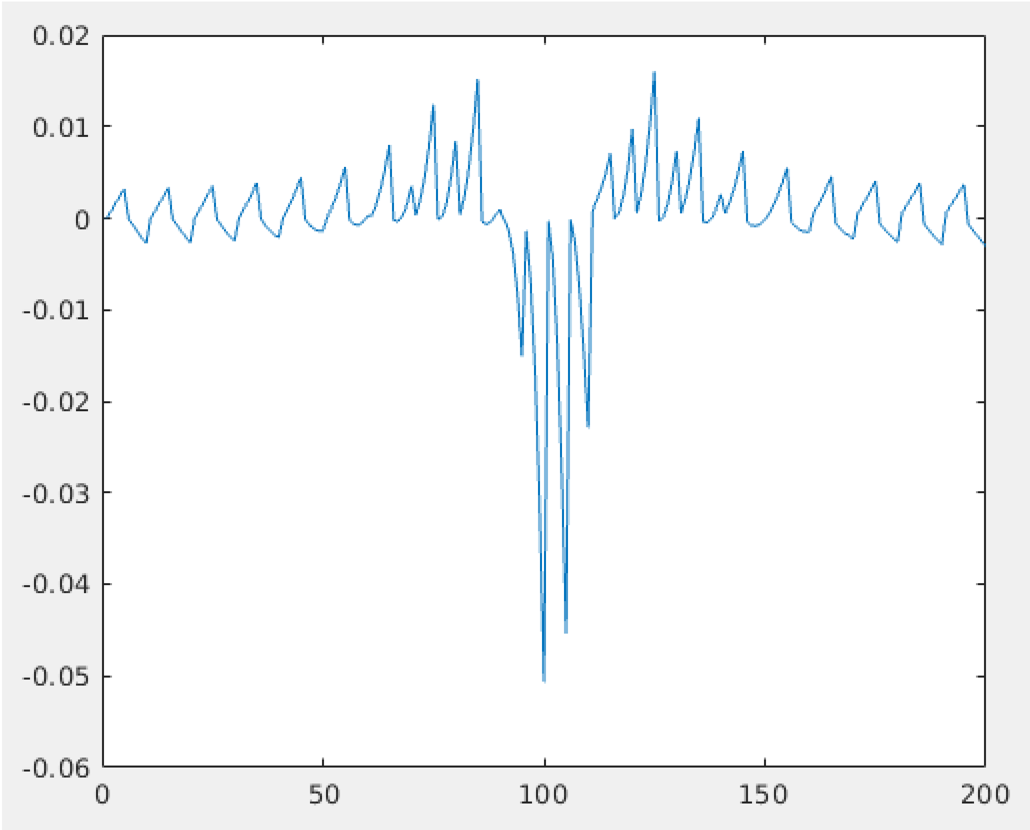
a)

To produce the following results open the matlab file Assignment6_6a.m and hit run. The SMSE will require you to copy the printed output back into the command line and hit enter to calculate the final value. The maximum negative and max positive require min(out) and max(out) to be put into the command line to get the following results.

SMSE = 0.007185064477276

Maximum positive error = 0.016012957999748

Maximum negative error = -0.050666041969688



Compare to Assignment 4 problem 8 the results here are within a much smaller range but the function of the difference between the values is not nearly as smooth.

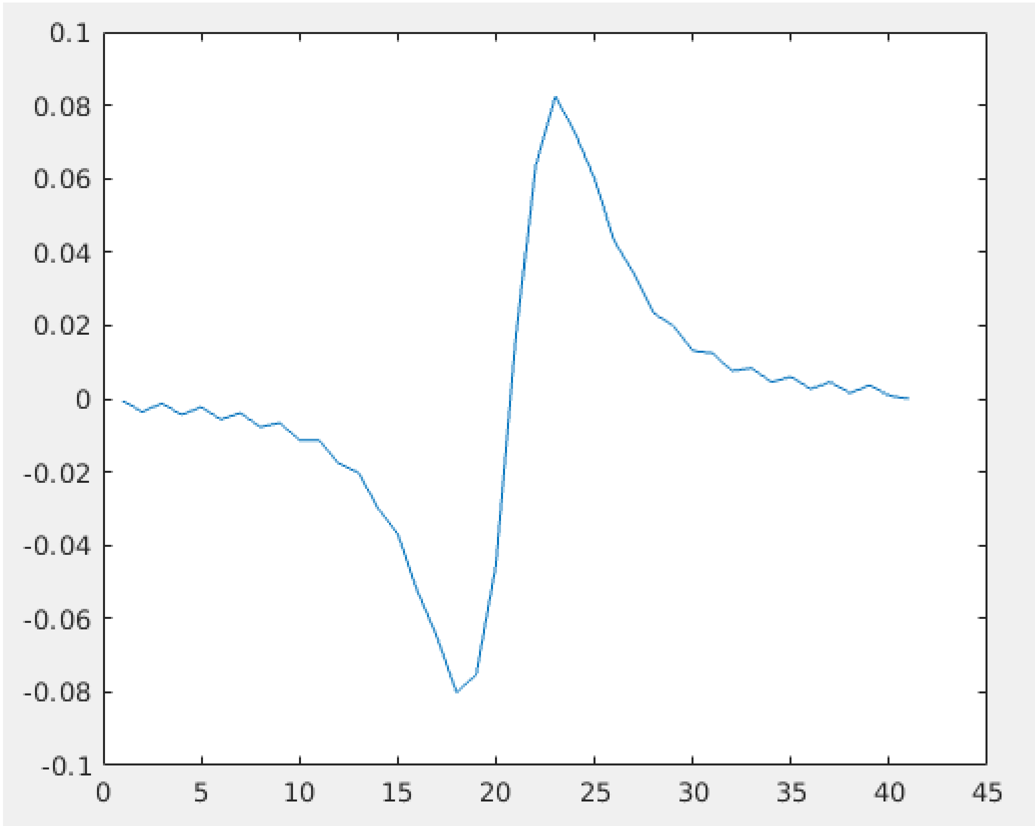
b)

To produce the following results open the file Assignment6_6b.m and hit run. To find the maximum negative and positive error values type max(y-z) and min(y-z) into the command line

SMSE = 0.034558911471291

Max = 0.082466196994828

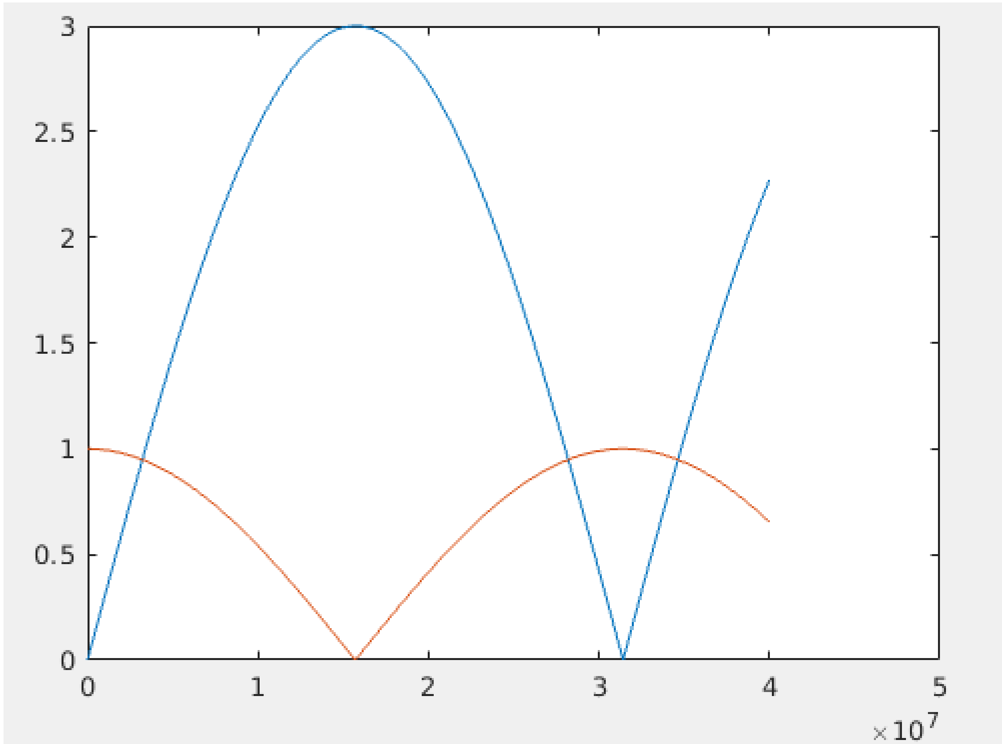
Min = -0.080113255818358



The following plot of the difference between values is even further refined to produce very little variance compared to the values produced in assignment 4 and even the previous problem. The function here is also represented by a smoother function indicating less variance in the values being produced which could mean a more exact method with less difference between the steps of the function.

Assignment 7 Problem 4

To reproduce the values and graph presented here, open the matlab file Assignment7_4.m and hit run. I was not able to produce a maximum error of 10^-5. The maximum step size I attempted was .0000001 and that is documented here. I am not sure if I can test much smaller of step size as to put that sort of load on my computer might be putting it's hardware in danger. Though the issue of not achieving the proper error level may have to do with the code itself as well. The maximum error seems to be stuck at around 3. Execution time required for this was 293.547407 seconds. The number of steps required were 1.570796300000000e+07.



Assignment 7 Problem 5

I was unfortunately unable to solve this problem.