README

Note that material for this assignment is divided between the python scripts in this folder as well as the scanned copy of my hand written answers to the more 'mathy' questions (apc523_hw1_mathanswers.pdf). Enjoy!

Question 2

Part A

244.71

Part B

Power Series Output: [244.71]
Iterations Required: 18
Relative Error: [mpf('-7.383870654188337e-5')]

Part C

0 [1.]
1 ['6.50%']
2 ['21.62%']
3 ['49.35%']
4 ['87.48%']
5 ['129.42%']
6 ['167.88%']
7 ['198.08%']
8 ['218.84%']
9 ['231.54%']
10 ['238.52%']
11 ['242.02%']
12 ['243.60%']
13 ['244.28%']
14 ['244.56%']
15 ['244.66%']
16 ['244.68%']
17 ['244.70%']
18 ['244.70%']
Power Series Output: [244.70]

Iterations Required: 18
Relative Error: [mpf('-3.2970992157163716e-5')]

Part D

Using exp() function: 0.00408677143846407
Section Iterations Final Value Error

Part d i 26 [0.0038363] [mpf('0.061288340254771249')]
Part d ii 20 [0.004] [mpf('0.021232270943118334')]
Part d iii 18 [0.] [mpf('1.0')]
Part d iv 19 [0.01] [mpf('-1.446919326422041')]

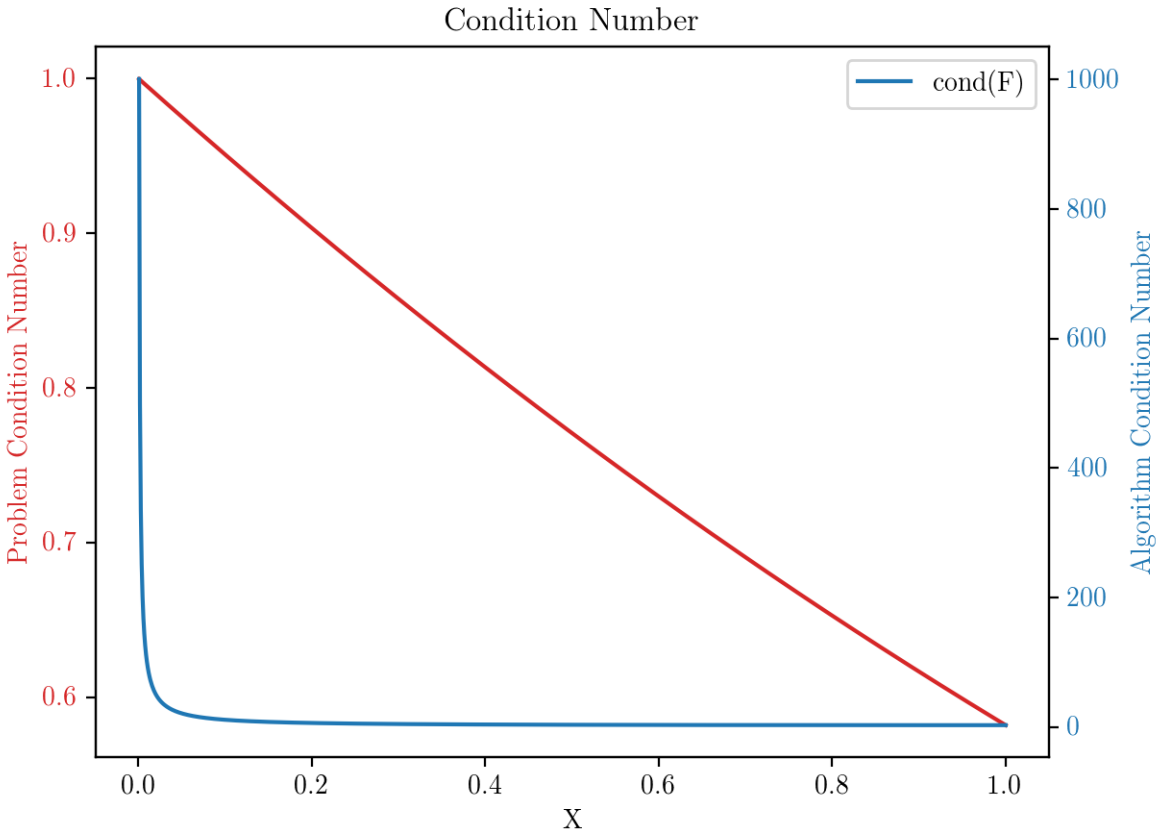
-option 3 converves the fastest but with 100% error
-option 2 has the lowest error (same as with positive x value)

Part E

Improve accuracy by expressing exp(-x) as 1/exp(x). IE calculate exp(x) and then invert the result. As shown adding right to left is still better.

method iterations output error
Adding Left to Right 19 [0.00408664] [mpf('7.3833254789675113e-5')]
Adding Right to Left 19 [0.00408664] [mpf('3.2969905106607524e-5')]

Question 4

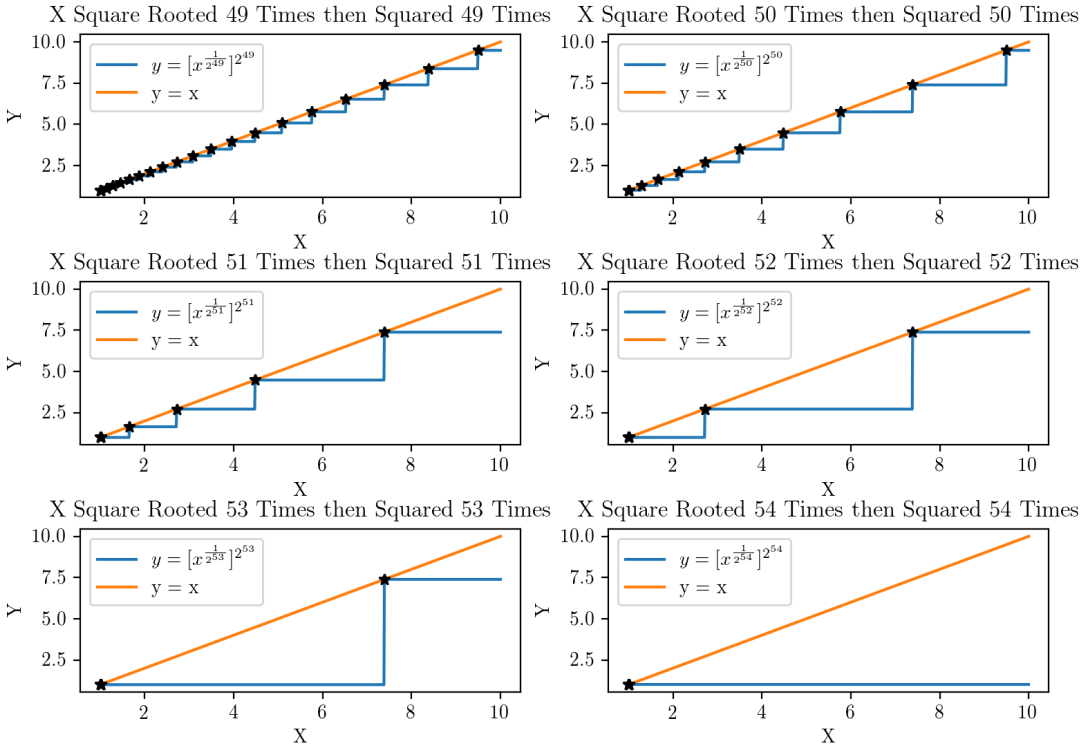

Condition Number
cond(F)
Problem Condition Number
Algorithm Condition Number
X
1.0
0.9
0.8
0.7
0.6
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0

Question 5

N gets large enough such that 1 + 1/n -> 1 since 1/n gets outside the range of the mantissa. Note that 10^-16 = 2^-53 which is the limit for double precision and it is after iteration 16 that the result goes to 1.

| i | N | e[i] | e[i-1]
2 [10.] ['2.59374246010%'] ['2.00000000000%']
3 [100.] ['2.70481382942%'] ['2.59374246010%']
4 [1000.] ['2.71692393224%'] ['2.70481382942%']
5 [10000.] ['2.71814592682%'] ['2.71692393224%']
6 [100000.] ['2.71826823719%'] ['2.71814592682%']
7 [1000000.] ['2.71828046910%'] ['2.71826823719%']
8 [10000000.] ['2.71828169413%'] ['2.71828046910%']
9 [1.e+08] ['2.71828179835%'] ['2.71828169413%']
10 [1.e+09] ['2.71828205201%'] ['2.71828179835%']
11 [1.e+10] ['2.71828205323%'] ['2.71828205201%']
12 [1.e+11] ['2.71828205336%'] ['2.71828205323%']
13 [1.e+12] ['2.71852349604%'] ['2.71828205336%']
14 [1.e+13] ['2.71611003409%'] ['2.71852349604%']
15 [1.e+14] ['2.71611003409%'] ['2.71611003409%']
16 [1.e+15] ['3.03503520655%'] ['2.71611003409%']
17 [1.e+16] ['1.00000000000%'] ['3.03503520655%']
18 [1.e+17] ['1.00000000000%'] ['1.00000000000%']

Question 6



See pdf for discussion.

Question 7

Part A

Coefficients:
[1.00000000000000 -210.000000000000 20615.0000000000 -1256850.00000000
 53327946.0000000 -1672280820.00000 40171771630.0000 -756111184500.000
 11310276995381.0 -135585182899530. 1.30753501054040e+15
 -1.01422998655115e+16 6.30308120992949e+16 -3.11333643161391e+17
 1.20664780378037e+18 -3.59997951794761e+18 8.03781182264505e+18
 -1.28709312451510e+19 1.38037597536407e+19 -8.75294803676160e+18
 2.43290200817664e+18]

Part B

Newton-Raphson: 20.0000241590258
Poly1d.roots: 19.999853539577177

Part C

Note some of these were complex and cast as real.

Coeff Value Newton root()
1.01 , [5.46959257] , [38.47818362]
1.0001 , [5.96933384] , [28.40021242]
1.000001 , [7.75268036] , [23.14901604]
1.00000001 , [9.58513982] , [20.64758289]

Part D

Coeff value Root 16 Root 17

-210.000000119209 , (16.73074488733208-2.8126248969405703j) , (16.73074488733208+2.8126248969405703j)

Roots 16 and 17 become complex conjugates.

Part E

Condition Numbers for:

| Root 14 | Root 16 | Root 17 | Root 20 |

58832087593943.3 | 38544752105144.2 | 17232905485151.3 | 137982741390.238