Programmer: Michael Lankford

Course: CSCI 3321 Section #1

Date: September 9,2022

Assignment: Number 1

Environment: Windows 10 Version 21H2 for x64-based Systems

Files Included: Error.java

ErrorTest.java

Purpose: Illustrate the effects of rounding errors and truncation errors  
  
 Input: emptyToFillArray- array that will be filled with all halved h values  
 h- initial value to be put into array and halved  
 inputNum- used for given x = 2 radians  
 filledArray- array of h values that will be plugged into derivative formula  
 derivArray- array of derived values that will be used to calculate error  
  
 Preconditions: Input value for the derivative is entered correctly  
  
 Output: Columns of correct i, h, approx, and error values  
  
 Postconditions: Correct table of values is outputted accordingly  
  
 Algorithm:  
 Import power method  
  
 Initialize emptyToFillArray  
 Initialize h  
 Initialize inputNum  
 Initialize filledArray  
 Initialize derivArray  
  
 Initialize emptyToFillArray = array for class instance  
 Initialize h = hIn for class instance  
  
 Initialize inputNum = input for class instance  
 Initialize filledArray = array for class instance  
  
 Initialize derivArray = input for class instance  
  
 fillArray method:  
 Start loop  
 Put new value into emptyToFill array  
 Divide h by 2 for new value  
 End loop  
 return emptyToFill array  
  
 derivative method:  
 Initialize output to filledArray size  
 Start loop  
 Compute f(x+h)  
 Compute f(x)  
 Initialize output element to f'(x) = (f(x+h) - (f(x)) / h  
 End loop  
 Return output array  
  
 error method:  
 Initialize err to derivArray size  
 Start loop  
 Initialize err element = derivArray element - (4 - 8cos(8))  
 End loop  
 return err array

Output:

i h approx error

0 1.000000e+00 6.740345e+00 1.576345e+00

1 5.000000e-01 6.611360e+00 1.447360e+00

2 2.500000e-01 1.078498e+01 5.620983e+00

3 1.250000e-01 8.972273e+00 3.808273e+00

4 6.250000e-02 7.192079e+00 2.028078e+00

5 3.125000e-02 6.191545e+00 1.027544e+00

6 1.562500e-02 5.679005e+00 5.150048e-01

7 7.812500e-03 5.421550e+00 2.575494e-01

8 3.906250e-03 5.292755e+00 1.287543e-01

9 1.953125e-03 5.228368e+00 6.436808e-02

10 9.765625e-04 5.196182e+00 3.218128e-02

11 4.882813e-04 5.180090e+00 1.608989e-02

12 2.441406e-04 5.172045e+00 8.044748e-03

13 1.220703e-04 5.168023e+00 4.022324e-03

14 6.103516e-05 5.166011e+00 2.011149e-03

15 3.051758e-05 5.165006e+00 1.005572e-03

16 1.525879e-05 5.164503e+00 5.027850e-04

17 7.629395e-06 5.164252e+00 2.513923e-04

18 3.814697e-06 5.164126e+00 1.256961e-04

19 1.907349e-06 5.164063e+00 6.284816e-05

20 9.536743e-07 5.164032e+00 3.142417e-05

21 4.768372e-07 5.164016e+00 1.571276e-05

22 2.384186e-07 5.164008e+00 7.857057e-06

23 1.192093e-07 5.164004e+00 3.928738e-06

24 5.960464e-08 5.164002e+00 1.961785e-06

25 2.980232e-08 5.164001e+00 9.857586e-07

26 1.490116e-08 5.164001e+00 4.791191e-07

27 7.450581e-09 5.164001e+00 2.407005e-07

28 3.725290e-09 5.164000e+00 1.214912e-07

29 1.862645e-09 5.164000e+00 2.281946e-09

30 9.313226e-10 5.164001e+00 2.407005e-07

31 4.656613e-10 5.164001e+00 2.407005e-07

32 2.328306e-10 5.164001e+00 1.194375e-06

33 1.164153e-10 5.164001e+00 1.194375e-06

34 5.820766e-11 5.164001e+00 1.194375e-06

35 2.910383e-11 5.164001e+00 1.194375e-06

36 1.455192e-11 5.164001e+00 1.194375e-06

37 7.275958e-12 5.164001e+00 1.194375e-06

38 3.637979e-12 5.164063e+00 6.222953e-05

39 1.818989e-12 5.164063e+00 6.222953e-05

Process finished with exit code 0

Conclusions: From this homework, I can conclude that as the h value for the derivative decreases, the calculations get closer to 5.1640002704689082070, which means it becomes more accurate.