CMSC 455/655 Homework 1
Dr. Tyler Simon Fall 2024

Name:

Date:

Please upload this document with your solutions as a PDF to blackboard.

Also upload any code that you wrote. Copy and paste code within the problem to show your methodology. Please comment appropriately. For instance, if you write two lines of code to calculate n!, then copy and paste those lines of code. Highlight and show the answer.

If you used source code (and you can for problems where it does not specify to write code explicitly), please cite the source code according to best practices.

Example: If you used the SPICEYPY library from NAIF,

Acton, C.H.; "Ancillary Data Services of NASA's Navigation and Ancillary Information Facility;" Planetary and Space Science, Vol. 44, No. 1, pp. 65-70, 1996.
DOI 10.1016/0032-0633(95)00107-7.

If you use my code, reference the website where I posted it.

Question 1: (10 points) Not all numbers are represented in base 2, and that is why we need to worry about floating point representation. Write the following programs and provide source code examples in your answer.

a) (3 points) Write a program to output the base 2 binary string of a decimal number with n digits using division and multiplication methods. Do not use built-in conversion functions. Ex: Input 3.14 will produce the binary string 11.0100011110

b) (2 points) Convert 38.1 out to 10 digits. Print the output.

c) (3 points) Write another program to convert a string binary number to floating point without using built-in functions. Using the 10-digit binary value for 0.1, convert back to base 10. What is the new value? Use as many digits as possible. What is the value using 52 digits in the conversion from base 10 to base 2?

d) (2 points) What is the relative and absolute error of the true value, 0.1, with the 10-digit rounded approximation? Repeat for 20 digits.

Question 2: (10 points) Write a program to add 0.001 to itself 1000 times. The exact analytical answer would be 1.0. What is the:

a) (2 points) Absolute error in your result with the true value?

b) (2 points) The relative error in your result with the true value?

c) (2 points) Why does the computer differ from the exact value?

d) (2 points) Using the code you developed in question 1, convert $0.1|_{10}$ to base-2 using 5 digits. Convert back, and then add that value to itself 10 times. Adding 0.1 to itself 10 times should equal the value of 1. What is the actual value? The absolute error?

e) (2 points) For the number of digits ranging from 5 to 100, plot the relative error in the calculated value and the true value. Place the plot below.

Question 3: (10 points) Horner's method reduces the number of operations necessary for evaluating n-degree polynomials. At large values of n, Horner's method, while fast, can produce large absolute errors. Write a program that uses Horner's method to evaluate any polynomial. Then for the following:

a) (3 points) Evaluate the polynomial $4x^5 + 7x^4 + 8x^3 -20x^2 -5x + 10$ normally and with Horner's method over the interval of [-2,2]. Plot the results using both. Label all results and axes.

b) (4 points) Evaluate the polynomial $x^9 -7x^7 +12x^5 + 90x^4 + 4x^3 -2x^2 +8x + 10$ normally and with Horner's method. Plot the results using both. Label all results and axes.

c) (3 points) Given the two polynomials, is there any difference between their output using Horner's method vs the normal method? If so, what is the absolute error? And the relative error? Plot the results using both. Label all results and axes.

Question 4: (10 points) On your computer, what are the largest and smallest numbers for double and single precision? What are the respective gaps between the smallest numbers and zero?

Question 5: (5 points) Copy and paste output showing the epsilon of your machine. State the coding language you are using.

Question 6: (15 points) Derive the Taylor series expansion for cos(2x) by hand. Show all the steps.

Question 7: (10 points) Write a short code without using built-in functions to calculate the Taylor series expansion of sin(x) out to n-number of terms.

a) (3 points) Plot the results of the Taylor series using 3 terms against the built-in function for sin(x) expanded around a = 0.

b) (2 points) Plot the results of the Taylor series using 10 terms against the built-in function for sin(x) expanded around a = 0.

c) (3 points) Derive a Taylor series expansion about a=1.2. Code this into a function and plot against the analytical function. Show the derivation by hand and plot for N=4 and N=6.

d) (2 points) What is the truncation error at a=1.2 for N=4 and N=10?

Question 8: (10 points) Create a code that uses bisection to find the roots of a curve. Then create a code that uses Newton's method. Within the code, keep track of the number of iterations used for each method to find each root.

a. The quadratic (x-0.3)*(x-0.5) has zeros at 0.3 and 0.5. Show that your bisection code can find both roots. How many iterations did your code use at each root? Be sure to mention what you used for your tolerance value.

b. Find the zeros using your code for Newton's method. How many iterations did it use for each root? Compare to the number of iterations used for bisection method.

c. Using your bisection code where do the curves of y=cos(x) and $y=x^3-1$ intersect?

Question 9: Calculate the FLOPS rate of your computer then write a program to measure the actual performance of your computer. Turn in your performance results and program.