EEE583-NUMERICAL ANALYSIS I

Midterm Exam 1-TAKEHOME PART

This is due three weeks. (Due date: Friday, April 15th, 2022).
Submit your work to me as a hard copy.
If you do not submit your work within three weeks, it will be graded to a zero.
This takehome part will be 50% of overall first midterm grade.

1-(10p)

a)(2p) What is "machine epsilon"?

b)(2p) What is the value of "machine epsilon" in Matlab (IEEE double-precision format in which eight bytes (64 bits) are used to represent floating-point numbers)

c)(6p) Compute the following calculations in Matlab:

```
x1 = 1+eps;
y1 = x1-1
x2 = 1+eps/2;
y2 = (x2-1)*2
```

What are the values of y1 and y2? Are those equal or not? (Note that they must be equal if you compute them using simple algebra.) Discuss with your reasoning.

2-(10p) Explore the use of the functions "round", "ceil", "floor" and "fix" in Matlab for the values x=0.3, x=1/3, x=0.5, x=1/2, x=1.65 and x=-1.34.

(This question helps understanding the role of various MATLAB commands which can be used to return different roundings to appropriate integers.)

3-(10p) The following code is supposed to evaluate the function for $x \in [0,1]$ (using 200 steps). Correct the code and check this by evaluating the function at $x = 1$ using $f(200)$ which should be -1/6.

$$f(x) = \frac{x^2 cos\pi x}{(x^3+1)(x+2)}$$

```
x = linspace(0,1);
clear all
g = x^3+1;
H = x+2;
z = x.^2;
y = cos xpi;
f = y*z/g*h
```

4-(10p) Calculate the "true errors" associated with the following calculations:

```
sin(15*pi)
(sqrt(2))^2
1000*0.001
1e10*1e-10
```

5-(20p)

a)(5p) Use **centered difference approximation** to estimate the **first derivative** of the following function at **x=0.5** using a step size **h=0.5.**

$$f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.25x + 1.2$$

b)(15p) Perform the same computation starting with **h=1**. Then progressively divide the step size by a factor of 10 to demonstrate how roundoff becomes dominant as the step size is reduced.

What is the **minimum value of h** that the minimum total error is reached?

6-(40p)

Find the root of the function $f(x) = e^x - e^{-2x} + 1$.

Perform the computation until ε_a is less than $\varepsilon_s = 0.01\%$.

a)(10p) Write a program using **bisection algorithm** and run your program on the function $f(x)$ with starting interval [-2,2]. How many steps (iterations) does the program use to achieve this tolerance?

b)(10p) Write a program using **fixed-point algorithm** and run your program on the function $f(x)$ with starting point -2. How many steps (iterations) does the program use to achieve this tolerance?

c)(10p) Write a program using **Newton-Raphson algorithm** and run your program on the function $f(x)$ with starting point -2. How many steps (iterations) does the program use to achieve this tolerance?

d)(10p) Write a program using **Secant algorithm** and run your program on the function $f(x)$ with starting interval [-2,2]. How many steps (iterations) does the program use to achieve this tolerance?

(Note that you need to **write your codes** and **submit them in your report** for each algorithm that you used.)