Numerical Methods (ME EN 2450)
Lab Midterm Practice, Fall 2024

Rules (for the actual in-lab exam)

- You must use a workstation in the CADE lab for the exam.
- The lab TA will answer questions regarding clarification of a problem description, but are not permitted to provide additional assistance.
- The exam is closed book and closed notes. You may not open a web browser nor access any previously-written files during the exam.
- If necessary, you may refer to the help documentation within Matlab or Spyder (for Python). You may also refer to the Matlab and Python notes provided for this exam.
- You can use built-in Matlab or Python functions to double-check your solutions, but utilizing these will not count toward your grade.
- The lab TA will tell you if you have the correct numerical solution, for each problem. If any are incorrect, you can retry the problem(s) once before submitting your exam.
- Before leaving the exam you must upload your code file(s) to Gradescope and verify the upload with your TA. You will not be submitting a report and uploads after you leave will not be permitted.

1 Fundamental Programming Concepts

In the following exercises, your code must generate the requested values based on n, i, and j. You can not hard-code the values and instead should rely on `for` loops and control structures, such as `if` and `else`.

a) Using a for loop, write a code to define the following vector:

$$b = \begin{bmatrix} 0 \\ 3 \\ 5 \end{bmatrix}$$

b) Using a nested for loop, write a code to define the following matrix:

$$A = \begin{bmatrix} 2 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 6 \end{bmatrix}$$

c) Using a nested for loop, write a code to compute the matrix-vector product:

$$[A] \cdot [b]$$

d) (+1 Extra Credit) Write a code to compute the same matrix-vector product, this time by replacing the inner for loop with the colon, [:], notation. You can use a built-in `sum` function for this.

2 Root Finding Methods in Engineering

Turbulent air flow in a pipe can be modeled using the Colebrook equation:

$$g(f) = \frac{1}{\sqrt{f}} + 2.0 \cdot log\left(\frac{\epsilon}{3.7D} + \frac{2.51}{Re\sqrt{f}}\right), \tag{1}$$

where f is the friction factor and Re is the Reynolds number, which is given by:

$$Re = \frac{\rho V D}{\mu} \tag{2}$$

Furthermore, the first derivative of the Colebrook equation is:

$$\frac{dg}{df} = -\frac{1}{2} \cdot f^{-3/2}\left(1.0 + \left(\frac{2.18261}{Re}\right) \cdot \left(\frac{\epsilon/D}{3.7} + \frac{2.51}{Re \cdot \sqrt{f}}\right)^{-1}\right) \tag{3}$$

Use the following for the physical constants:

- $\epsilon = 1.43x10^{-6}$ [m]
- $\rho = 1.38$ [kg/m^3]
- $\mu = 1.91x10^{-5}$ [Ns/m^2]
- $D = 0.0061$ [m]
- $V = 54$ [m/s]

Objectives:

a) Plot Eqn. 1 to graphically approximate the root, $g(f) = 0$.

b) Write a code to approximate the root using the Bisection method.

c) With an initial guess (plot from part (a)) and code (part (b)) determine the root of Eqn. 1 to within 0.1% approximate normalized error and report the required number of iterations.

d) Reproduce the plot from part (a), but now with the approximate root from each iteration included.

As a reminder, the fundamental equation for the Bisection method is:

$$c_{i+1} = \frac{a+b}{2} \tag{4}$$

where c_{i+1} is the next approximate root, a is the lower bracket bound, and b is the upper bracket bound.

3 Linear Systems in Engineering

You are working on a construction project and need 4800, 5810, and 5690 m^3 of sand, fine gravel, and coarse gravel, respectively.
There are 3 pits from which you may extract materials. The composition of materials in each pit is given in the following table:

	Pit 1	Pit 2	Pit 3
Sand %	52	20	25
Fine Gravel %	30	50	20
Coarse Gravel %	18	30	55

How many cubic meters must be hauled from each pit to meet the requirement?

a) Write a system of 3 equations and 3 unknowns (i.e. the percent to be extracted from each pit).

b) Solve for the unknowns using a Gauss elimination code that you write (pseudocode provided below).

c) Assume that the material composition in each pit remains fixed, but the requirements of sand, fine gravel, and coarse gravel changes for every project. Would LU decomposition be a better solution method than Gauss elimination to accommodate future projects? Explain.

Gauss Elimination Pseudocode

Forward elimination

```
for k = 1 to n − 1 do
    for i = k + 1 to n do
        s = a_ik / a_kk
        for j = k to n do
            a_ij = a_ij − s a_kj
        end
        b_i = b_i − s b_k
    end
end
```

Backward substitution

```
x_n = b_n / a_nn
for i = n − 1 to 1 by −1 do
    s = 0
    for j = i + 1 to n do
        s = s + a_ij x_j
    end
    x_i = (b_i − s) / a_ii
end
```