# ME EN 2450 Assignment HW 7

Name: ______________________________

______________________________

You must sumbit this assignment to **gradescope**.

I declare that the assignment here submitted is original except for source material explicitly acknowledged.

I also acknowledge that I am aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the University website.

| | |
|---|---|
| Name | Date |
| Signature | Student ID |

## Score

| | |
|---|---|
| Q1 | ____ /12 |
| Q2 | ____ /10 |
| Total: | ____ /22 |
| Extra credit | /2 |

## Q1. Eigenvalues and eigenvectors of a matrix

Consider the following matrix

$$\begin{bmatrix} 2 & 8 & 10 \\ 8 & 4 & 5 \\ 10 & 5 & 7 \end{bmatrix}$$

(a) (2 points) To find eigenvalues of this matrix using the polynomial method, derive the characteristic equation (i.e., the equation that takes the form of polynomial = 0).

(b) (2 points) To solve the characteristic equation, use **a root finding code (any method) you previously wrote**. You need to find at least one root (i.e., one eigenvalue for the linear system). Submit both your code and your result.

(c) (2 point) Find all eigenvalues using **the same code** and clearly explain how you achieved that.

(d) (4 points) **Independently**, write your own code to use the Power Method to determine both the largest and the smallest eigenvalues of the matrix.
Carry out 5 iterations for each eigenvalue.
Calculate the approximate relative error at each iteration.
Note that you code should be able to handle a general square matrix of any size
(Hint 01: Refer to Examples 27.7 and 27.8 in the textbook.).
(Hint 02: You can use built-in functions in Matlab or Python to find the matrix inverse.)

(e) (2 points) Ask ChatGPT or similar Artificial Intelligence (AI) tools available online (BingChat, Bard, Claude, etc) to write a piece of code for you with the same requirements specified in (d).
Test the code and write a short (1 to 3 sentences) compare and contrast between your own code and code generated by the AI tool.

(f) (Extra Credit 2 points) Try to improve your own code so that it is at least better than the AI code in one aspect (any aspect would be fine).
Clearly define this particular aspect you choose in 1 sentence and then explain why your code is better.
*NOTE: Dr. Pai and the TA team will not answer any questions regarding this extra credit task.*

Please also study **Lecture18_notes_PowerMethod.pdf** in the Lecture folder before you attempt this question.

NOTE: The largest/smallest eigenvalues are defined in terms of their magnitudes regardless of the positive or negative sign.

## Q2: ODE eigenvalue problem

An axially loaded wood column (simply supported on both ends) has the following characteristics:

- $E = 10 \, \text{x} \, 10^9 [Pa]$
- $I = 1.25 \text{ x } 10^{-5} [m^4]$
- $L = 3[m]$

$$P = \pi^2 \frac{EI}{L^2} \,, \tag{1}$$

where $P$ is the analytical solution for the critical buckling load.

Reference Equations 27.17, 27.18, 27.20, and Examples 27.7 and 27.8 of the textbook for this question.

1. (1 point) Calculate the analytical buckling load of the first mode (n = 1) using Equation 1.
2. (2 points) By coding, use Power Method in Matlab or Python which takes two inputs: the matrix and the number of iterations.
   The code should return the **smallest** eigenvalue. Submit your code.
   (Hint 01: You can use the same code as Q1, either the AI-generated one or your own one.)
   (Hint 02: You can use built-in functions in Matlab or Python to find the matrix inverse.)
3. (3 points) Using finite differences [see Equation 27.18 of the textbook], set up the coefficient matrix that results from using 5 nodes (2 boundary nodes and 3 interior nodes), evenly distributed along the column. By calling your Power Method function, compute the buckling load after 1, 2, 3, 4, and 5 iterations. Submit your code, the tabulated results of the numerically-approximated buckling load vs. Power Method iterations.
4. (4 points) Increase BOTH the level of discretization (i.e., putting more nodes along the column) AND the number of iterations in Power Method, such that the numerically-approximated buckling load is within 1% from the analytical value. (i.e., relative true error < 1%)

Please also study **Lecture18_notes_ColumnBuckling.pdf** in the Lecture folder before you attempt this question.