Homework 3

(Due: April 12, 2024)

Question 1: 10 points. Figure 1 shows an elastic column fixed at its base and pinned at the top. The critical buckling load, P_{cr}, corresponds to the first positive solution to

$$\lambda = \tan[\lambda] \tag{1}$$

where $P_{cr} = \lambda^2 \left[\frac{EI}{L^2}\right]$.

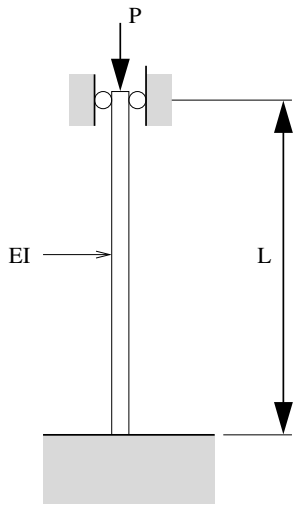

Figure 1: Elastic column carrying an axial load.

Use Python to create a single plot of y_1 = x and y_2 = tan(x) – the intersection of contours on this plot should give you an approximate range within which an accurate solution exists. Use the **method of bisection** to compute the numerical solution to equation 1 Print the buckling load corresponding to your solution?

Question 2: 10 points. An efficient way of computing the cube root of a number `N` is to compute the root of

$$f(x) = x^3 - N = 0 \tag{2}$$

with the method of Newton Raphson, namely:

$$x_{n+1} = x_n - \left[\frac{f(x_n)}{f'(x_n)}\right] \tag{3}$$

Substituting equation 2 into equation 3 and rearranging terms gives the recursive relationship

$$x_{n+1} = \frac{1}{3} \cdot \left[\frac{2x_n^3 + N}{x_n^2}\right] \tag{4}$$

Write a test Python program that will compute the cube root of a number `N` via equation 4. The details of the cube root calculation should be contained within a method having the declaration:

```
def cuberoot(  N, tolerance, maxiterations ):
```

Your cube root method should use:

$$\left|\frac{x_{n+1} - x_n}{x_n}\right| < \varepsilon \tag{5}$$

for a test on convergence, where ε is a very small number (i.e., the tolerance). Print the number `N`, its cube root, whether or not the numerical procedure converged, and finally, the number of iterations. Verify that your test program works for a variety of positive and negative values of N (e.g., N = 8, 1000, -8, -27, etc).

Question 3: 10 points. The quartic function

$$f(x) = (x-3)^2(x-5)^2 \tag{6}$$

has double roots at x = 3 and x = 5. Write a Python program to plot f(x) over the range [2,6]. Demonstrate that while the method of **Newton Raphson** struggles to find value(s) of x for which f(x) = 0, the method of **Modified Newton Raphson** computes the double roots with ease.

Note. I suggest that you use the NewtonRaphson and Modified Newton Raphson code distributed in class.

Question 4: 10 points. Consider the family of matrix equations AX = B defined by

$$\begin{bmatrix} 1 & 0 & 1 \\ a & 8 & 4 \\ 0 & 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ b \end{bmatrix}. \tag{7}$$

Determine the values of 'a' for which matrix A will be singular. Then,

1. Develop a program that uses NumPy to store matrices A and B, and then systematically evaluates the determinant and rank of A, and the rank of augmented matrix [A | B] for values (a,b) corresponding to: (a) zero solutions, (b) infinite solutions.

2. Develop a second program that uses SymPy to store matrices A and B symbolically, and then computes symbolic solution to the matrix equations 7. For the values of (a,b) that make A singular, evaluate the determinant and rank of A, and the rank of augmented matrix [A | B].

You should find that the Python module SymPy is considerably more powerful than its numerical counterpart NumPy.

Question 5: 10 points. In the design of crane structures, engineers are often required to compute the maximum and minimum member forces and support reactions due to a variety of loading conditions.

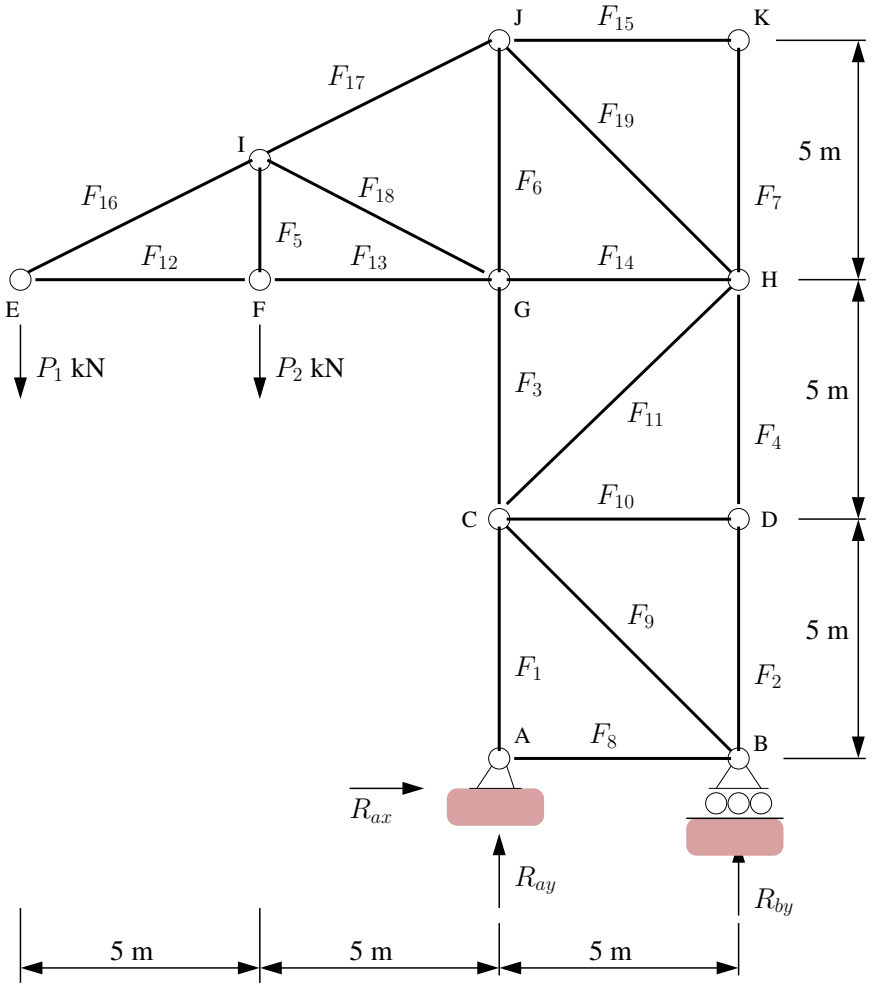

Figure 2: Front elevation of crane tower.

Figure 2 shows a nineteen bar pin-jointed truss carrying vertical loads P_1 kN and P_2 kN at joints `E` and `F`.

The symbols F_1, F_2, $\cdots$ F_{19} represent the axial forces in truss members 1 through 19, and R_{ax},R_{ay}, and R_{dy} are the horizontal and vertical support reactions at joints `A` and `B`.

Write down the equations of equilibrium for joints `A` through `K` and put the equations in matrix form. Now suppose that the crane supports a variety of payloads that, for engineering purposes, it can be represented by the sequence of external load vectors

$$\begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} 15 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} 10 \\ 10 \end{bmatrix}, \quad \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 25 \end{bmatrix} kN. \tag{8}$$

Develop a Python program that will solve the matrix equations for each of the external load conditions, and compute and print the minimum and maximum support reactions at nodes A and B, and axial forces in each of the truss members.