

## Assignment 8: Matlab Scripts and Plots

Date Due: March 28, 2024

Instructor: Trani

### Problem 1

An engineer would like to solve the system of linear equations:

$$23x_1 + 45x_2 + 67x_3 + 56x_4 + 22x_5 = 120$$

$$13x_1 + 35x_2 + 37x_3 + 26x_4 + 20x_5 = 130$$

$$18x_1 + 14x_2 + 45x_3 + 29x_4 + 16x_5 = 200$$

$$20x_1 + 14x_2 + 15x_3 + 19x_4 + 19x_5 = 210$$

$$17x_1 + 14x_2 + 45x_3 + 29x_4 + 16x_5 = 312$$

- A) Create a script in Matlab to solve the system of equations. You can read the data from a text file or just enter the information in the script. Display the results for each value of x into the Command Window with a nice format. For example :  $x_1 = 35$  and so on.

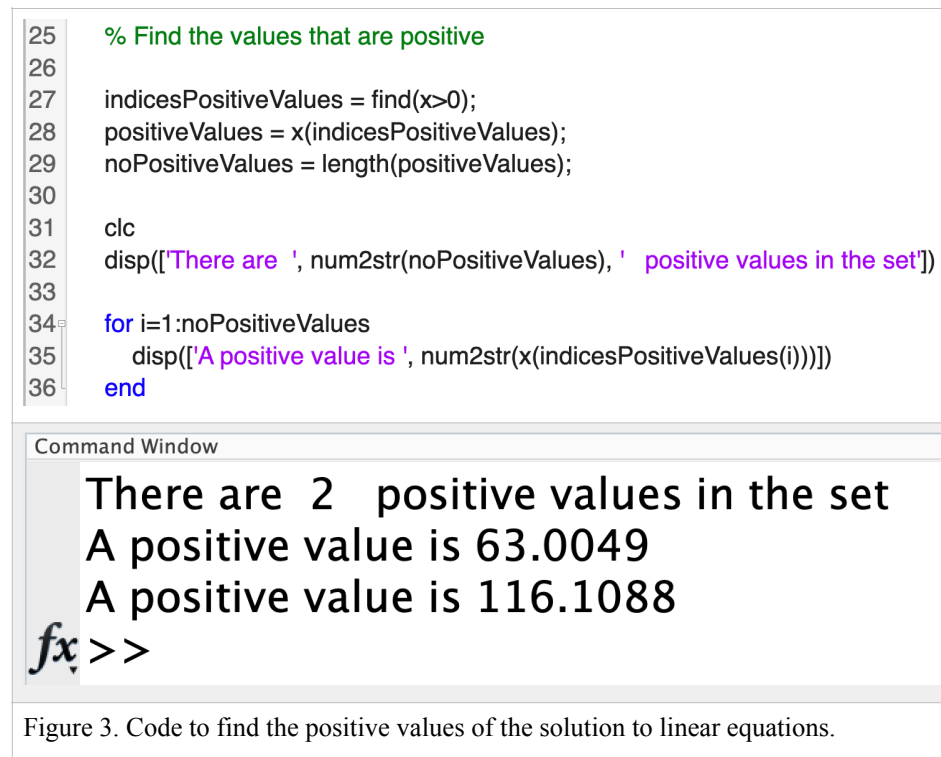
```
lp_sollution_a3_2010.m x solveEquations_A8_P1.m x +
1 % Script to solve a set of linear equations
2
3 % Define the RHS of the equations
4
5 A=[23 45 67 56 22
6     13 35 37 26 20
7     18 14 45 29 16
8     20 14 15 19 19
9     17 14 45 29 16];
10
11 b = [120 130 200 210 312]';
12
13 % Solve the equations
14
15 x=A\b;
16
17 % Display the results in the Comand window
18 clc
19 disp(['The value of x1 is ', num2str(x(1))])
20 disp(['The value of x2 is ', num2str(x(2))])
21 disp(['The value of x3 is ', num2str(x(3))])
22 disp(['The value of x4 is ', num2str(x(4))])
23 disp(['The value of x5 is ', num2str(x(5))])
```

Figure 1. Code to find the solutions to the system of linear equations.

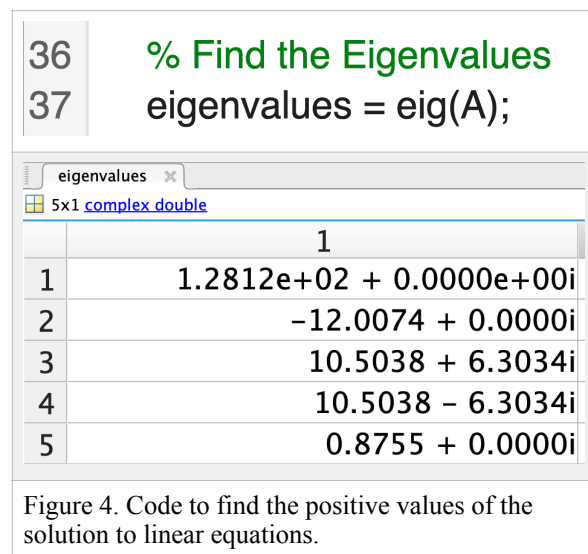
```
Command Window
The value of x1 is -112
The value of x2 is -49.6641
The value of x3 is -17.1907
The value of x4 is 63.0049
The value of x5 is 116.1088
```

Figure 2. Solution to the system of linear equations.

B) Add code to the script to find the values of  $x$  that are positive. Display those in the Command Window.



C) Find the Eigenvalues of the right-hand side matrix containing the coefficients of each  $x$  using the “eig” command in Matlab.



## Problem 2

A train engineer finds that the basic train resistance of a high-speed train (see Figure 1) to be:

$$R = A + BV + CV^2$$

$$P_r = 1000 * R * V$$

Where:

$R$  is the basic train resistance in kiloNewtons

$P_r$  is the power required to move the train in Watts

$V$  is the train speed in meters/second

$A$ ,  $B$ ,  $C$  are empirical constants derived from wind tunnel test. The values of  $A$ ,  $B$ ,  $C$  are 8.1, 0.11, and 0.012, respectively.

- A) **Create a Matlab script** to estimate the value of the basic train resistance ( $R$ ) and the power required ( $P$ ) as a function of train speed ( $V$ ). For this analysis create a vector of speeds between 0 and 80 meters/second at intervals of 1 m/s and estimate resistance and power.

```
calculate_trainresistance.m x [+]  
1 % Script to estimate train resistance and power  
2 % Equations  
3 %  
4 % R = A + B * V + C * V^2 (kiloNewtons)  
5 % P = 1000 * R * V; ( Watts)  
6 %  
7 % Define the values of A, B, and C  
8  
9 A = 8.1;  
10 B = 0.11;  
11 C = 0.012;  
  
13 % Create a vector with values of speed  
14 V = 0:1:80; % Speed (m/s)  
15  
16 % Calculate R and P  
17 R = A + B * V + C * V.^2;  
18 P = 1000 * R .* V;  
19  
20 % Plot the values of R and P  
21 figure  
22 plot(V,R,'or')  
23 xlabel('Train Speed (m/s)')  
24 ylabel('Train Resistance (kN)')
```

Figure 5. Code to estimate train resistance and power.

- B) Plot the train resistance (in Newtons) and the power required in Watts. Use a subplot to plot both graphs in a 2 by 1 matrix configuration.

```

calculate_trainresistance.m x +
20 % Plot the values of R and P
21 figure
22 subplot(2,1,1)
23 plot(V,R,'o-r')
24 xlabel('Train Speed (m/s)')
25 ylabel('Train Resistance (kN)')
26 grid
27
28 subplot(2,1,2)
29 plot(V,P,'o-b')
30 xlabel('Train Speed (m/s)')
31 ylabel('Power (Watts)')
32 grid

```

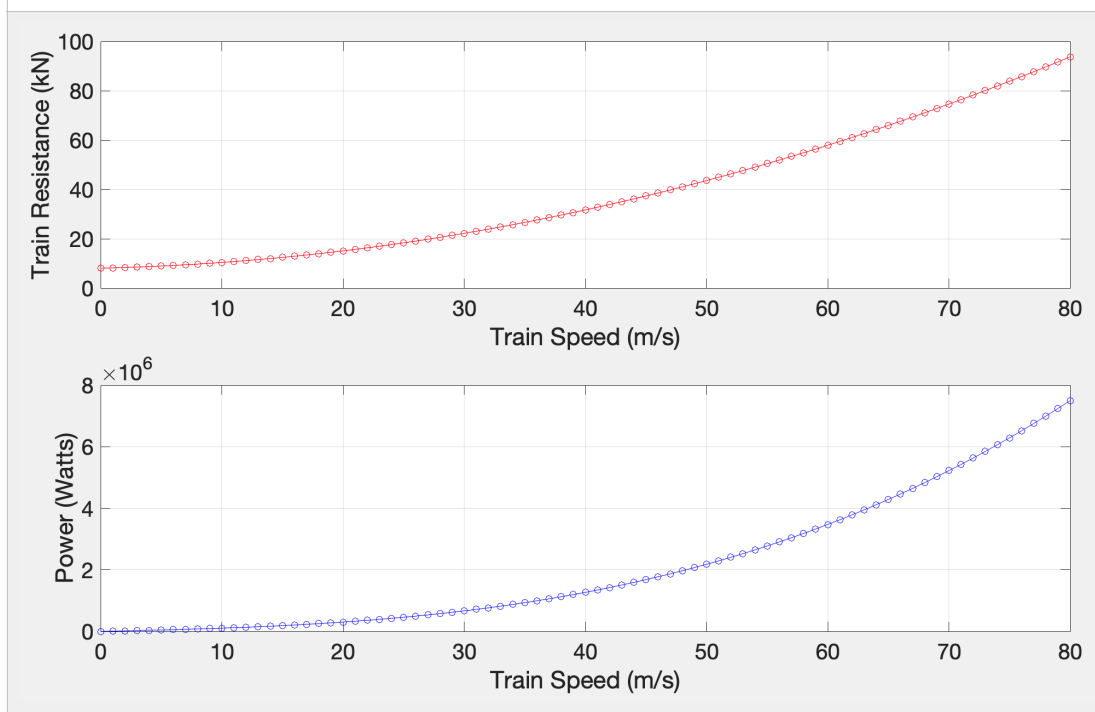


Figure 6. Code to plot the train resistance and power.

C) Convert the power required to horsepower in the script and plot horsepower (in y-axis) as a function of train speed (in the x-axis).

```
34 % Convert power to HP
35 watts2HP = 0.00134102;
36 P_hp = P * watts2HP;
37
38 plot(V,P_hp,'o-b')
39 xlabel('Train Speed (m/s)')
40 ylabel('Power (HP)')
41 grid
```

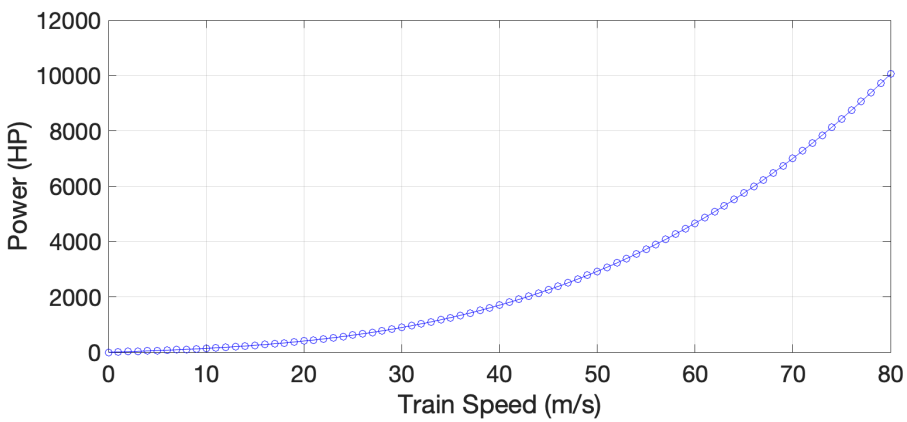


Figure 7. Code to plot the train power in HP.

### Problem 3

A simple formula used in highway engineering to estimate the horizontal radius of a road is:

$$R = \frac{v^2}{g(e + f)}$$

where:

$R$  is the road horizontal radius (in meters)

$v$  is the road design speed (m/s)

$g$  is the gravitational constant (9.81 m/s<sup>2</sup>)

$e$  is the superelevation rate of the road (%/100). For example, a road with a superelevation rate of 0.06 implies the road of inclined at a lateral slope of 0.06 meters for each one meter in horizontal distance (see . Typical values of  $e$  vary from 0 (no superelevation) to 0.10 (high superelevation).

$f$  is the lateral friction force coefficient developed between the vehicle tires and the pavement (see Figure 2).

The lateral friction force coefficient varies according to the table below. This means, at higher speeds, the car has less grip on the road to execute a turn.

Road Design Speed (m/s)	Lateral Friction Coefficient $f$ (dim)
5.0	0.18
8.3	0.17
12.5	0.16
16.7	0.15
25.0	0.12
28.0	0.11
33.4	0.09
35.0	0.08

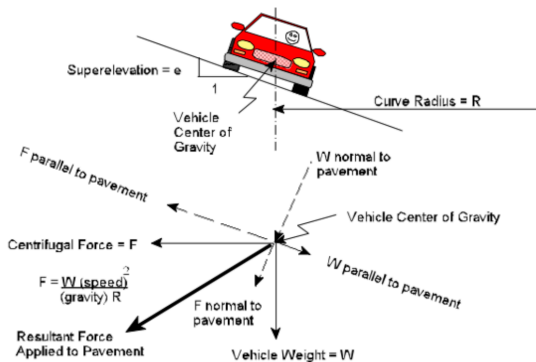


Figure 2. Road Superelevation Diagram. NYDOT.

- A) Plot the road design speed versus the lateral friction coefficient in Matlab (see values in Table 1). Use the **Basic fitting method** demonstrated in class to find the best **second-order polynomial** of the data.

The objective is to find the value of lateral friction coefficient ( $f$ ) for any speed required. The estimation of  $f$  can be performed using the linear regression approximation of  $f$  versus speed ( $v$ ). The regression equation is of the form:

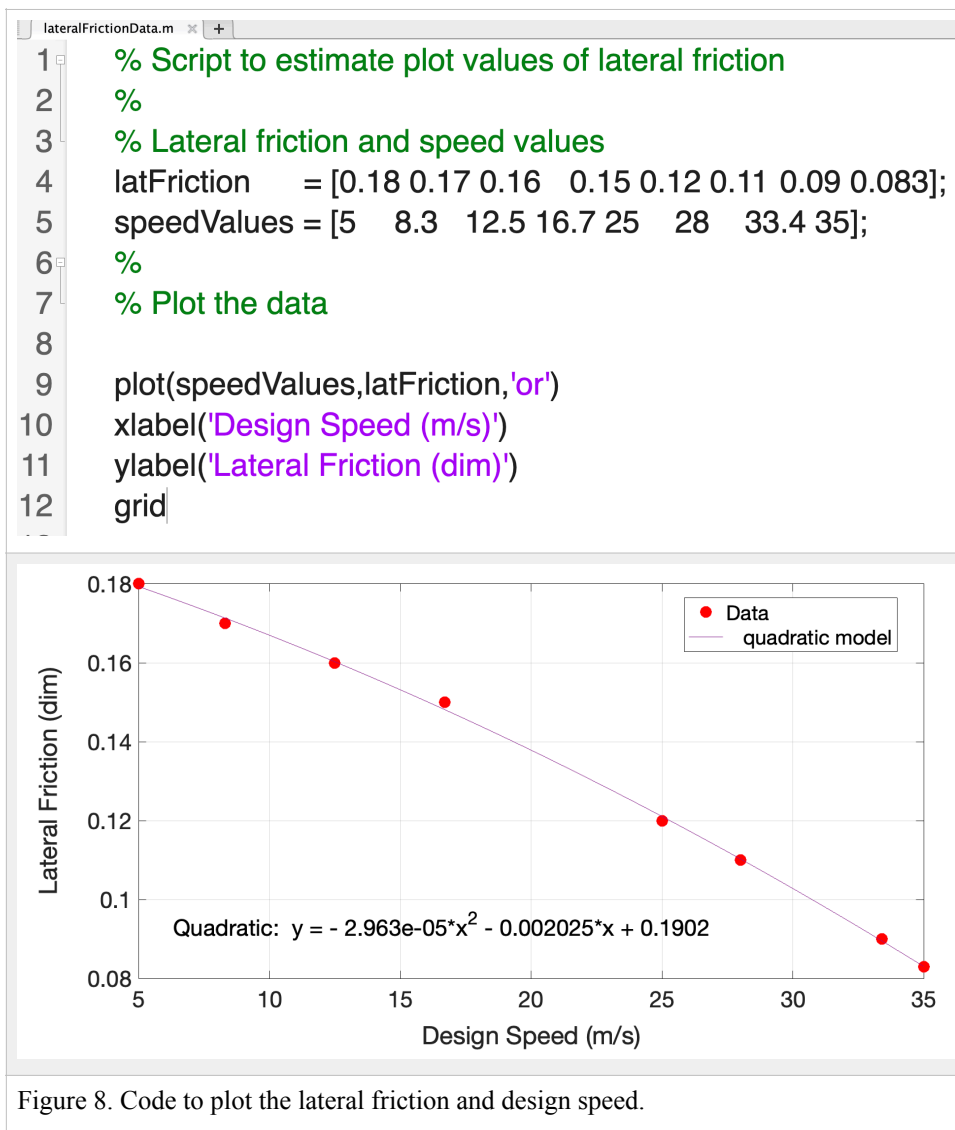
$$f = A + Bv + Cv^2$$

where:

$A, B, C$  are the regression coefficients to be found using the Matlab Basic fitting method.

$v$  is the road design speed (m/s).

Show me the plot and the equation of the second order polynomial found using Matlab.



Coefficients are:  $C = -2.96276e-05$ ,  $B = -0.0020248$  and  $A = 0.190205$



- B) Create a Matlab script to estimate the radius of the horizontal curve (in meters) according to the equation provided as a function of design speed. Find the solutions for speed values ranging from 10 to 35 m/s (steps 05 m/s). Use a constant value of superelevation rate 0.04.

```

lateralFrictionData.m
14 % Find the radius of the curve as a function of design speed
15 % Define superelevation and gravity
16 superelevation =0.04; % dimensionless
17 g = 9.81; % gravity constant (m/s-s)
18
19 % Polynomial fit to calculate the lateral friction coefficient
20
21 C = -2.962760e-05;
22 B = -0.0020248;
23 A = 0.1902052;
24
25 designSpeed = 10:1:35; % design speed in meters per second
26 lateralFrictionCoeff = A + B * designSpeed + C * designSpeed.^2;
27
28 Radius_m = designSpeed.^2 ./ (g * (lateralFrictionCoeff + superelevation));
29
30 % Make a plot
31 plot(designSpeed,Radius_m,'o-b|)
32 xlabel('Design Speed (m/s)')
33 ylabel('Radius of the Curve (m)')
34 grid

```

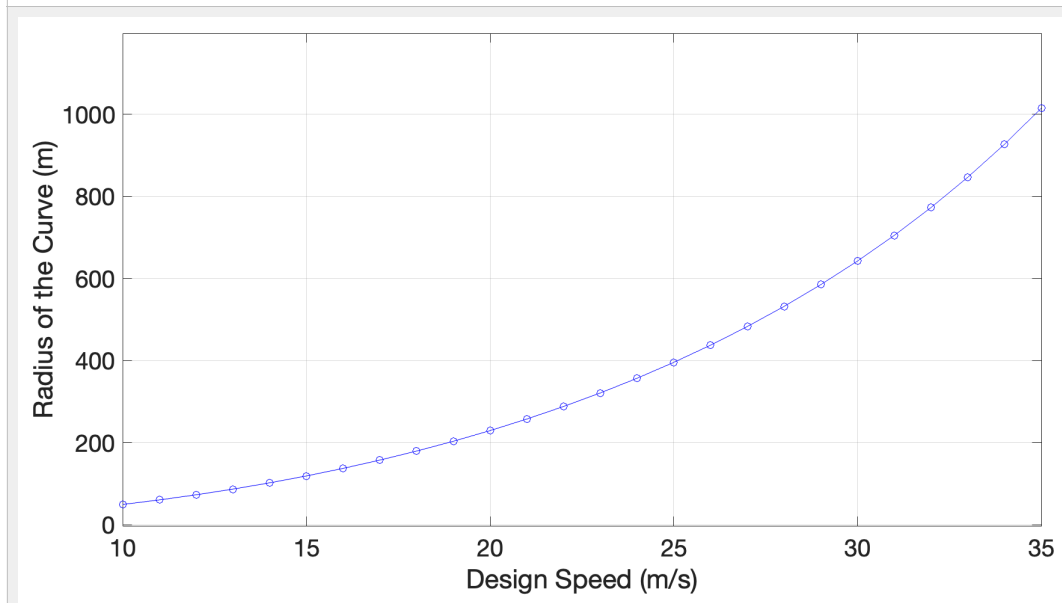


Figure 9. Code to plot the radius of the curve.

- C) Plot the radius of the curve for the speed values and label accordingly.  
D) Repeat the process for superelevation rate of 0.08.

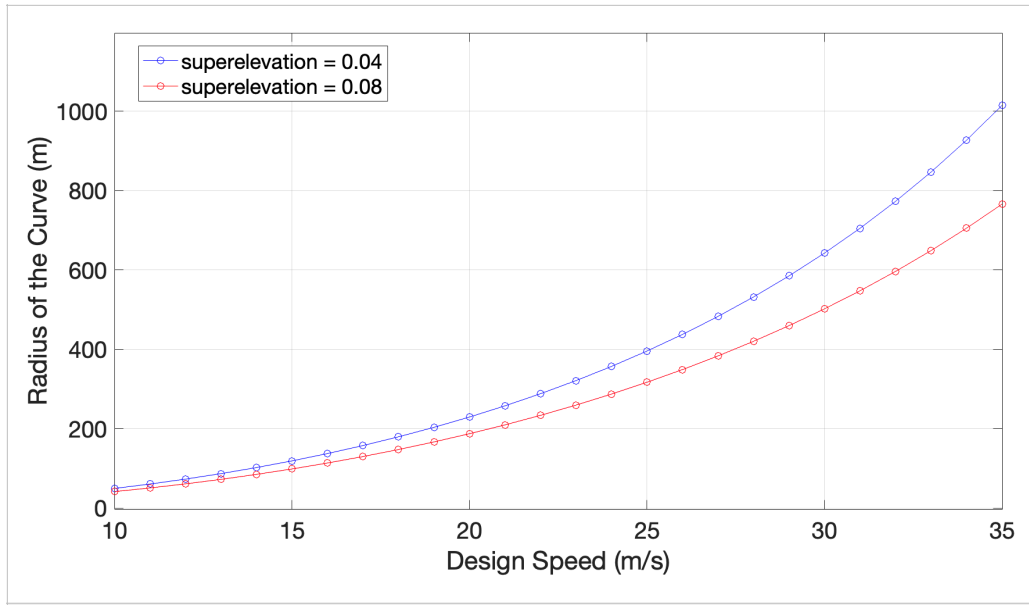


Figure 10. Plot the radius of the curve for two superelevations.

E) Comment on the effect of superelevation in the radius of the curve. How could superelevation help the civil engineer?

**The higher value of superelevation reduces the radius of the curve. This may reduce the cost of construction of a road or highway.**

## Problem 4

Figure 11 shows a beam supported on both ends with a single load applied at a distance “a” and “b” from both ends.

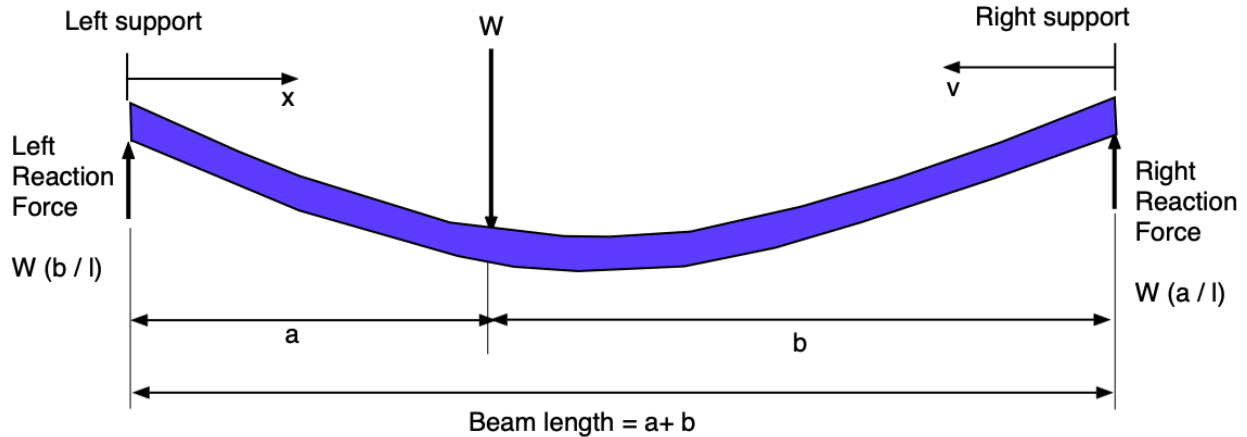


Figure 11. A simple beam supported at both ends with a load  $W$  at some known location. Adapted from: [http://www.engineersedge.com/beam\\_bending/beam\\_bending3.htm](http://www.engineersedge.com/beam_bending/beam_bending3.htm). The nomenclature for beam deflection equations are shown below.

$W$  = load (N)

$E$  = Modulus of elasticity ( $\text{N/m}^2$ )

$I$  = Moment of inertia ( $\text{m}^4$ )

$x$  = Beam station (or distance) from left side of the beam to the loading point (m)

$v$  = Beam station (or distance) from right hand side beam end point to the loading point (m)

$l$  = beam length (m)

$a, b$  = distances from each beam end point towards the loading point (m)

Let:

$$y_a = \frac{-Wbx}{6EI}(l^2 - x^2 - b^2)$$

$$y_b = \frac{-Wav}{6EI}(l^2 - v^2 - a^2)$$

where:

$y_a$  = deflection of the beam from left support point (section a) in meters

$y_b$  = deflection of the beam from right hand side support point (section b) in meters

### Task 1

Create a **Matlab script** to estimate the deflections of the beam ( $y_a$  and  $y_b$ ). Inputs to the script are the parameters  $W, E, I, a, b$  and vectors  $x$  and  $v$ . The output of the Matlab script are two vectors with values of deflection ( $y_a$ ) and ( $y_b$ ) for the given stations  $x$  and  $v$ . Your function should be able to handle vector operations.

```

1 % Script to calculate the deflection (y) of a cantilever beam
2 % subject to a single load (W)
3 % A. Trani (2024)
4
5 % W = load at station x (N)
6 % E = Modulus of elasticity (N/m-m)
7 % I = Moment of inertia (m-m-m-m)
8 % x = beam station = distance from datum point (wall) to any point on the
9 % beam (m)
10 % l = beam length (m)
11 % y = beam deflection at any station (m)
12 % a = distance between load and left side
13 % b = distance between load and right side
14
15 % Beam properties
16
17 W = 3900; % Newtons
18 E = 205e9; % N/m-m - value for Steel = 200e9
19 I = 0.0008; % meters to the fourth power
20 a = 3.0; % meters
21 b = 3.75; % meters
22 length_of_beam = a + b; % meters
23 x = linspace(0,a,300); % 300 points to the end of the beam - every 0.01 meters
24 v = linspace(0,b,5); % 5 points to the end of the beam - every 0.01 meters
25
26 % Calculate deflection to the beam at any point in the beam before the load
27 % W (distance a)
28
29 ya = -W .* x * b / (6*E * I * length_of_beam) .* (length_of_beam^2 ...
30 -x.^2 - b^2);
31
32 yb = -W .* v * a / (6*E * I * length_of_beam) .* (length_of_beam^2 ...
33 -v.^2 - a^2);
34
35 % Plot the deflection
36
37 % figure
38 subplot(2,1,1)
39 plot(x, ya, 'b')
40 xlabel('Station (meters)')
41 ylabel('Deflection (meters)')
42 grid
43
44 subplot(2,1,2)
45 plot(v, yb, 'b')
46 xlabel('Station (meters)')
47 ylabel('Deflection (meters)')
48 grid
49
50 % Second plot with distances re-arranged
51
52 v_rev = length_of_beam - v;
53
54 figure
55 plot(x, ya, 'b', v_rev, yb, 'r')
56 xlabel('Station (meters)')
57 ylabel('Deflection (meters)')
58 grid

```

Figure 12. Code to estimate the deflection of the beam.

## Task 2

Test your Matlab script with the values shown below. Clearly state the values of the deflections at the values of the beam stations ( $x$  and  $v$ ) shown below.

$W = 3900 \text{ N}$   
 $E = 205e9 \text{ N/m}^2$   
 $I = 0.0008 \text{ m}^4$   
 $b = 3.75 \text{ meters}$   
 $a = 3.00 \text{ meters}$

Find the deflection of the beam ( $y_a$ ) for values of  $x$  ranging from 0 to "a" with step size 0.01 meters. Also, find the deflections of the beam at stations  $v$  from 0 to "b" with step size 0.01 meters.

In the calculations for Task 2 define  $x$  and  $v$  as a vector to produce two output vectors of deflections ( $y_a$ ) and ( $y_b$ ).

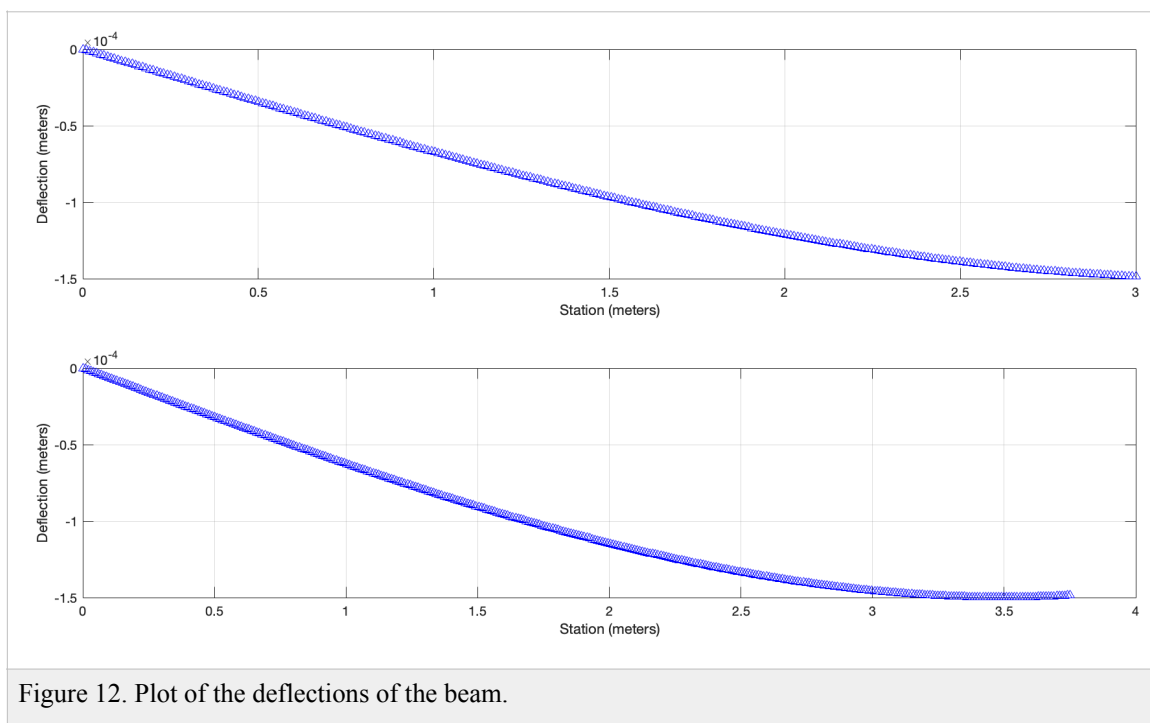


Figure 12. Plot of the deflections of the beam.

## Task 3

Make a subplot of the beam deflection ( $y_a$ ) as a function of beam station ( $x$ ). Make a subplot of the beam deflection ( $y_b$ ) as a function of beam station ( $v$ ). Arrange the subplots in a single figure with deflections  $y_a$  in the top plot.

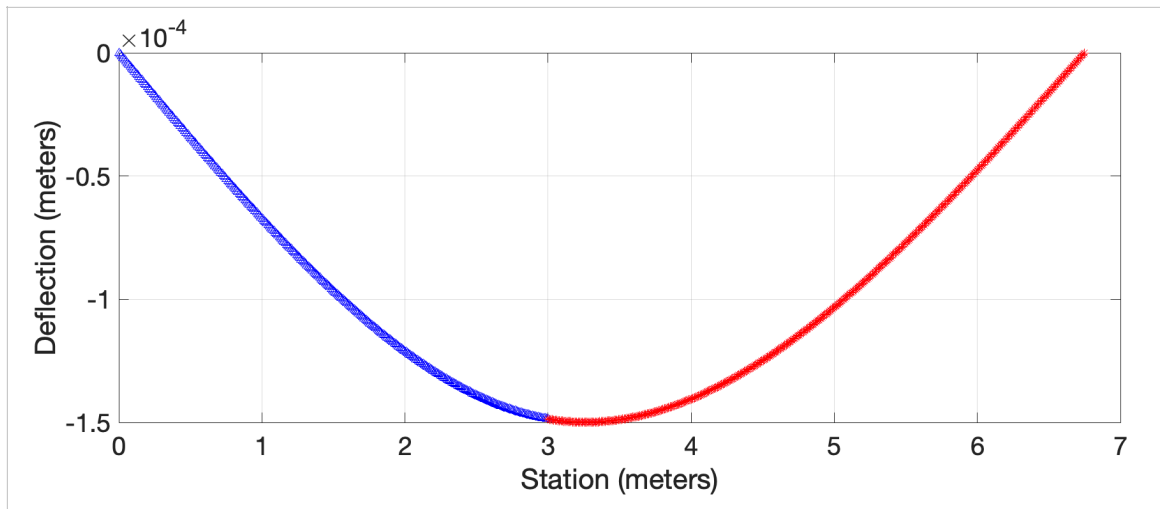


Figure 12. Plot of the deflection of the beam.

#### Task 4

Add code to the script created in tasks 1-3 to estimate two reaction forces and to measure the CPU it takes to do the calculations in your computer. Display in the Command Window the reaction forces and the CPU time to do the calculations.

**The CPU time is 0.15024 seconds**