Assignment 7: Basic Matlab Operations

Solution Instructor: Trani

Problem 1

Use Matlab to solve the problem.

A common problem in Civil Engineering is to estimate the deflection of a cantilever beam at different stations (see Figure 1).

$$y = \frac{Px^2}{6EI}(3l - x) \qquad \text{(Equation 1)}$$

Where:

y is the deflection at any point in the beam (inches)

x is the distance from the wall to any point on the beam (inches)

P is the load applied (lbs.)

l is the length of the beam (inches)

E is the modulus of elasticity (lb/in^2)

I is the moment of inertia (in^4)

The units in this model are all consistent.

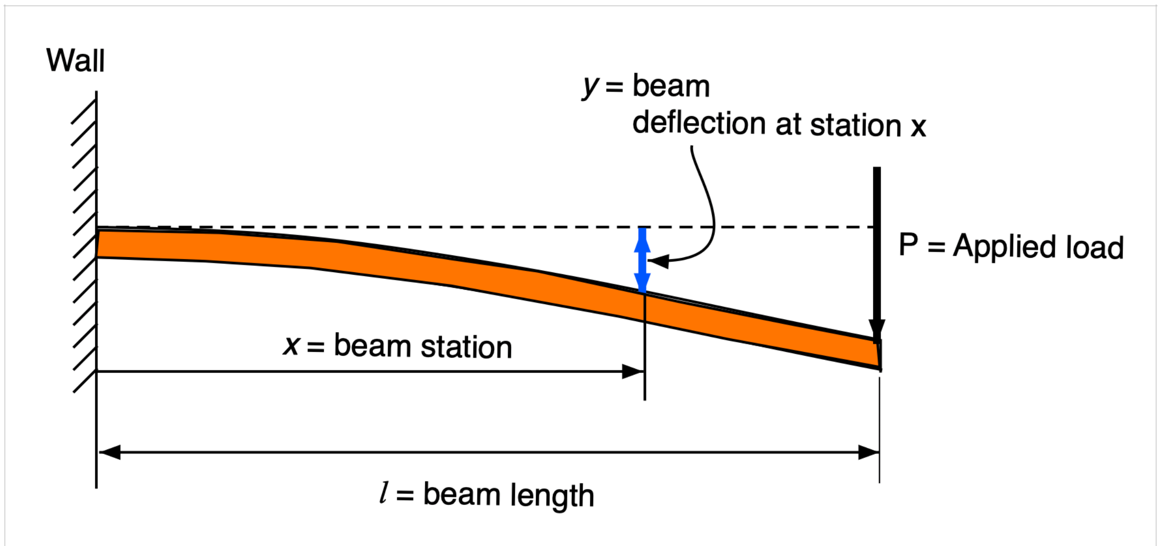

Figure 1. Cantilever Beam Subject to a Single Load (P).

A) Create a new Matlab script to estimate the estimate the beam deflection (y) given the parameters on the right hand side of Equation 1.

The values of the parameters are:

x = 0:1:250;   % vector of values defined in Matlab from the wall to the end of the beam (inches)

P = 2000 lbs.

l =250 inches

E = 2.9e7 (in^2)

I = 200 (in^4)

```
% Script to estimate the beam deflection
% subject to a single load (at the end of the beam)

% Inputs: P, L, x, E , I
% Output : deflection (y)

P = 2000;          % load in pounds
I = 200;           % inches^4
E = 2.9e7;         % modulus of eleasticity in inches^2
L = 250;           % inches (beam length)

x = 0:1:L;            % vector of stations from the wall

% Find the deflection of the beam

y = P .* x.^2 / (6 * E * I) .* (3 * L - x);

% Plot of the deflections versus station (x)

plot(x,y,'ob')
xlabel('Station (inches)')
ylabel('Deflection (inches)')
```

Figure 2. Matlab Script to Calculate the deflection of a Cantilever Beam Subject to a Single Load (P).

B) Plot the deflection (*y*) as a function of the beam station (*x*).

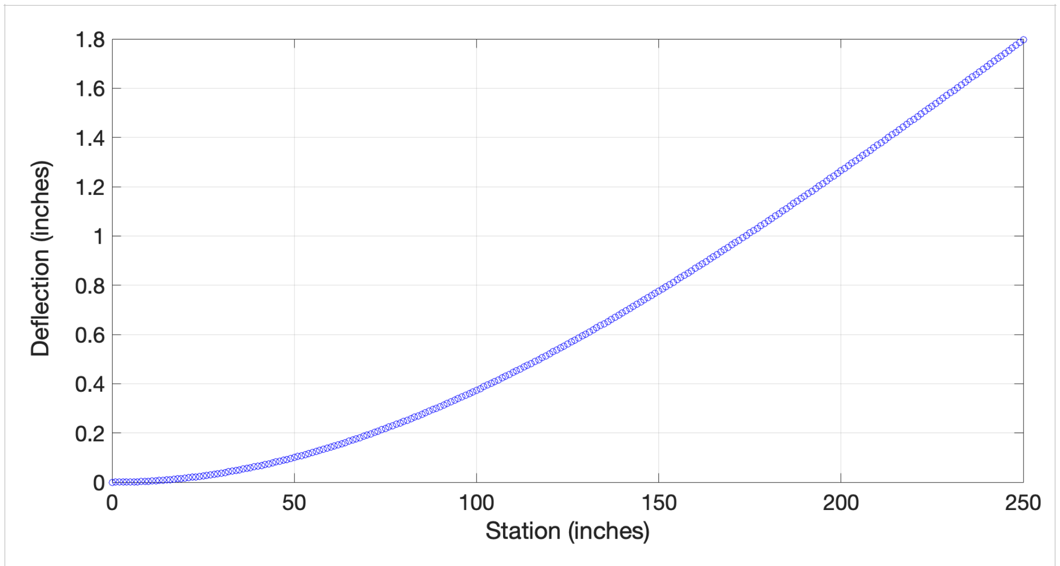

Figure 3. Cantilever Beam Deflection Subject to a Single Load (P).

C) Using the Matlab **max(y)** command, find the maximum deflection of the beam.

[i n]=max(y)

i = 1.7960 inches

n =251 (the last data point of the beam)

D) Modify the script created in part (A) creating another variable z where z is defined as:

$z = gradient(y)$

The function gradient takes the first derivative of the values of vector (y)

Using the "subplot" command create a new figure with two plots. Plot the values of x vs. y in the upper part of the window and x vs. in the lower part of the same window. Change the line colors to distinguish the two views.

d) Verify that the "gradient" function is working.

```
% Add code to get the gradient of the deflection

gradDeflection = gradient(y,x);

figure
subplot(2,1,1)
plot(x,y,'ob')
xlabel('Station (inches)')
ylabel('Deflection (inches)')
grid
```

```
subplot(2,1,2)
plot(x,gradDeflection,'or')
xlabel('Station (inches)')
ylabel('Gradient of Deflection (inch/inch)')
grid
```

Figure 4. Code to Calculate the Gradient of the Deflection.

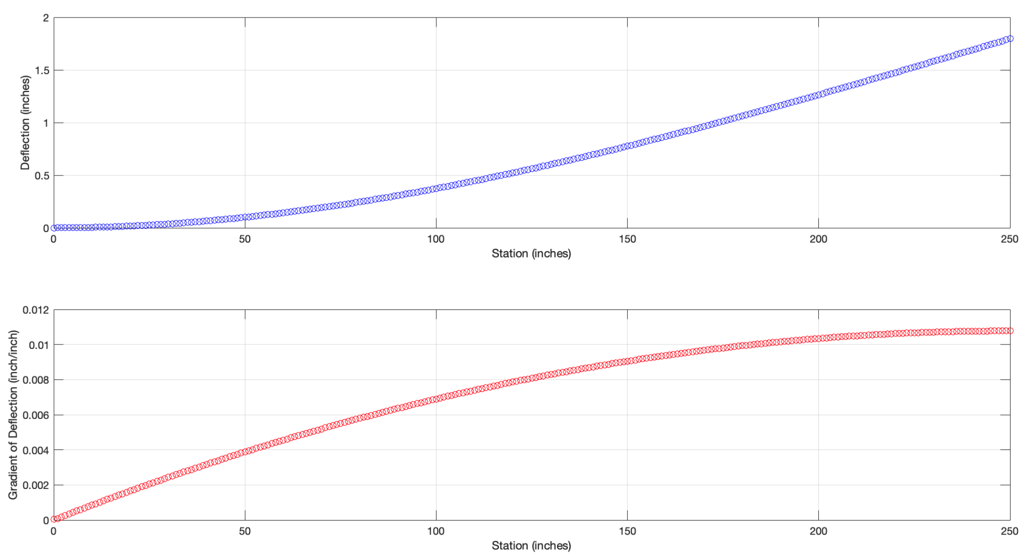

Figure 5. Plot of the Gradient of the Deflection.

Problem 2

Use the **US_dams_Units_2024.xlsx** file to work on the problem. For each problem show the Matlab code used to execute the task.

	A	B	C	D
1	Name	DamID	Longitude_deg	Latitude_deg
2	ANZALDUAS DIVERSION	TX83301	-98.334	26.137
3	DELTA LAKE UNIT NO 2 LEVEE	TX03790	-97.9367	26.43
4	FALCON	TX00024	-99.167	26.559
5	STRUCTURE 79	FL00310	-81.7126	26.7255
6	G-90	FL00440	-81.41	27.34
7	LAKE CASA BLANCA DAM	TX02267	-99.4483	27.5333
8	EXISTING STACK	FL10026	-82.38	27.87
9	EDWARD MEDARD RESERVOIR DAM	FL00434	-82.16	27.91
10	P-11	FL13003	-81.85	27.93

Figure 6. U.S. Dams.

a) Use the Matlab wizard to read the all data. Let **Matlab create the code** to read the data. Save the file.

b) Create individual variables for each column of data provided. Label the variables according to the headers in the Excel file.

readUS_Dam_Data.m

```
%% Import data from spreadsheet
% Script for importing data from the following spreadsheet:
%
%    Workbook: /Users/vuela-adm/Courses/CEE 3804/CEE 3804
%    Worksheet: damData
%
% Auto-generated by MATLAB on 30-Mar-2023 19:38:19

%% Set up the Import Options and import the data
opts = spreadsheetImportOptions("NumVariables", 13);
```

readUS_Dam_Data.m

```
opts.Sheet = "damData";
opts.DataRange = "A2:M5201";

% Specify column names and types
opts.VariableNames = ["Name", "DamID", "Longitude_deg", "Latitude_
opts.VariableTypes = ["string", "string", "double", "double", "string", "do

% Specify variable properties
opts = setvaropts(opts, ["Name", "DamID", "County", "Hazard", "State"
opts = setvaropts(opts, ["Name", "DamID", "County", "Hazard", "State"
```

readUS_Dam_Data.m

```
%% Convert to output type
Name = tbl.Name;
DamID = tbl.DamID;
Longitude_deg = tbl.Longitude_deg;
Latitude_deg = tbl.Latitude_deg;
County = tbl.County;
Height_ft = tbl.Height_ft;
MaxStorage_acreFoot = tbl.MaxStorage_acreFoot;
NormalStorage_acrefoot = tbl.NormalStorage_acrefoot;
SurfaceArea_acres = tbl.SurfaceArea_acres;
DrainArea_acres = tbl.DrainArea_acres;
```

Figure 7. Matlab Generated Code to Import Data.

c) Plot the longitude of the dam (x-axis) versus the latitude of the dam location (y-axis) to get an idea of the locations of the dams in the US.

readUS_Dam_Data.m

```
load usamap

hold on
plot(uslon,uslat,'-k')
xlabel('Longitude (deg)','fontsize',22)
ylabel('Latitude (deg)','fontsize',22)

% Add the US dam locations

plot(Longitude_deg,Latitude_deg,'ob')
```

Figure 8. Matlab Code to Plot the Data.

The tasks is plotted in part (d).

d) Use the supplied US map to plot over the plot created in part (c). Use the Matlab ‘hold on’ command to keep the plot created in part (c) active.

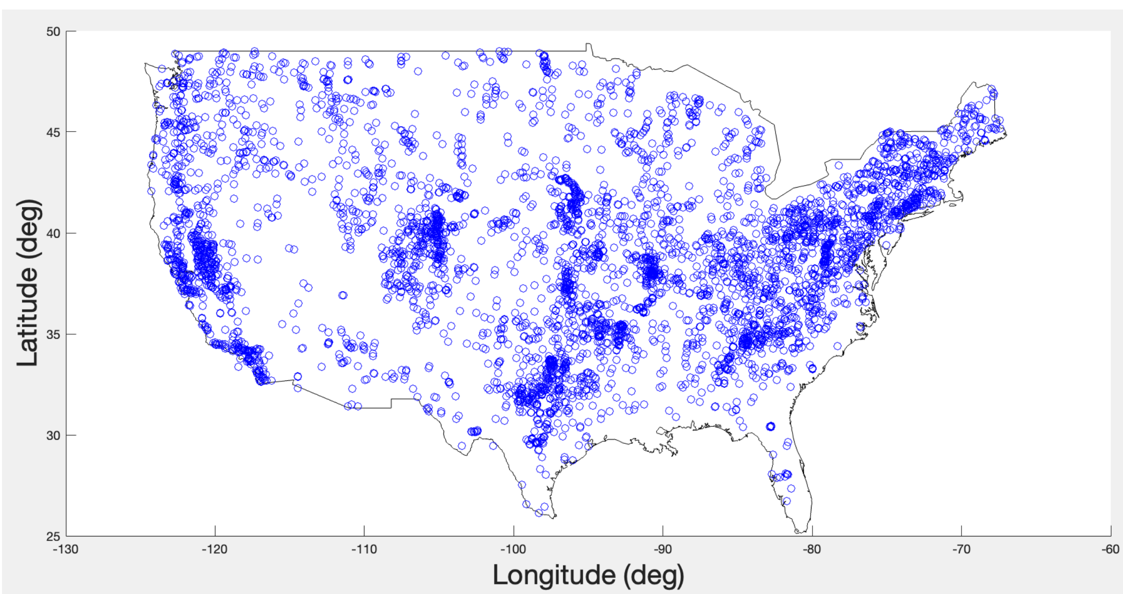

Figure 9. Plot of U.S. Dams.

e) Use the string comparison command (strcmp) to find the dams in the state of Virginia. Create a variable with the names of the dams in Virginia. Show me the name of the first 15 dams listed in your answer.

```
% Find the dams in Virginia
% Use the strcmp command first
% Then use the find commend to select values that match the state
matchesForVirginia = strcmp(State,'VA');
indicesForVirginia = find(matchesForVirginia);

damsInVirginia = Name(indicesForVirginia);
```

Figure 10. Code to Find the Dams in Virginia.

damsInVirginia

130x1 string

	1
1	ISLAND CREEK DAM
2	C-POND DAM
3	STEWARTS CREEK - LOVILLS CREEK DAM #9
4	SCHOOLFIELD DAM
5	JOHN H KERR DAM
6	HORSE PASTURE CREEK DAM #1C
7	CLEAR CREEK
8	TALBOTT
9	TOWNES
10	LEATHERWOOD CREEK DAM #5
11	LEATHERWOOD CREEK DAM #2A
12	LAKE COHOON DAM
13	BEAVER CREEK DAM
14	LAKE MEAD DAM

Figure 11. Names of the Dams in Virginia.

f) Add to the script to find the number of dams in Ohio.

```
% Find the dams in Ohio
% Use the strcmp command first
% Then use the find commend to select values that match the state
matchesForOhio = strcmp(State,'OH');
indicesForOhio = find(matchesForOhio);

numberOfDamsInOhio = length(indicesForOhio);

disp(['There are  ', num2str(numberOfDamsInOhio), '    dams in Ohio'])
```

Figure 12. Code to Find the Number of Dams in Ohio.

```
There are  128    dams in Ohio
>>
```

Problem 3

Use the GPS data collected by a car data logger to do this problem. A sample of the data is shown below.

	A	B
1	Time (s)	Speed (km/h)
2	0	0.00
3	2	10.00
4	4	22.00
5	6	35.00
6	8	48.00
7	10	62.00
8	12	63.00

Figure13. Car Speed Data.

a) Read the data using Matlab.

```
opts.VariableTypes = ["double", "double"];

% Import the data
tbl = readtable("/Users/vuela-adm/Courses/CEE 3804/CEE 3804

%% Convert to output type
Times = tbl.Times;
Speedkmh = tbl.Speedkmh;
```

Figure13. Partial Code of Matlab Generated Code to Read the Car Data.

b) Plot the car speed (in y-axis) vs. time (x- axis). Observe the plot and comment on the number of stops.

```
figure
plot(Times,Speedkmh,'o-r')
xlabel('Time (seconds)')
ylabel('Speed (km/hr)')
```

Figure14. Code to Plot the Car Data.

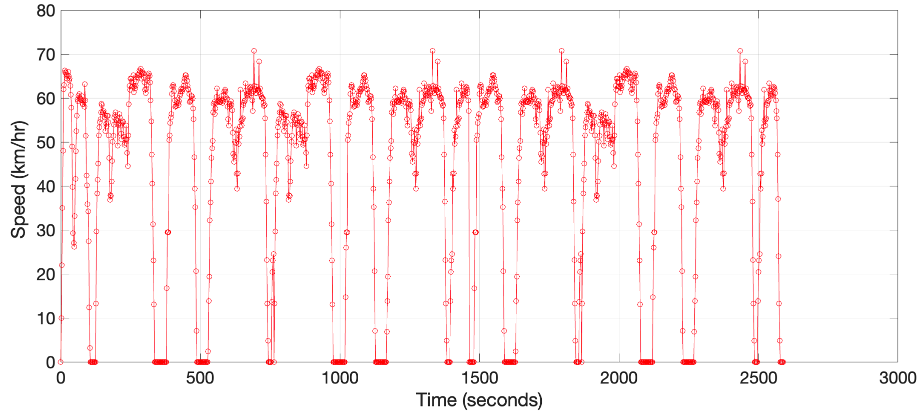

Figure 15. Plot of the Car Data.

c) Convert the speed data into units of meters per second. Create a new variable to store the speed values in m/s.

```
% New variable called Speedms
Speedms = Speedkmh / 3.6;                    % in meters per second
acceleration = gradient (Speedms,Times);     % in m/s-s
```

Figure16. Code to Convert to Meters per Second and to Calculate Car Acceleration.

d) Convert the speed data into units of miles per hour and create a new variable.

e) Estimate the acceleration of the car as a function of time using the metric data. Use the Matlab “gradient(x)” function to find the acceleration using the speed vector created in part (c). Plot the calculated acceleration vs. time recorded by the GPS data logger unit.

```
% Plot acceleration versus time
figure
plot(Times,acceleration ,'o-b')
xlabel('Time (seconds)')
ylabel('Acceleration (m/s^2)')
grid
```

Figure 17. Code to Plot Car Acceleration.

f)

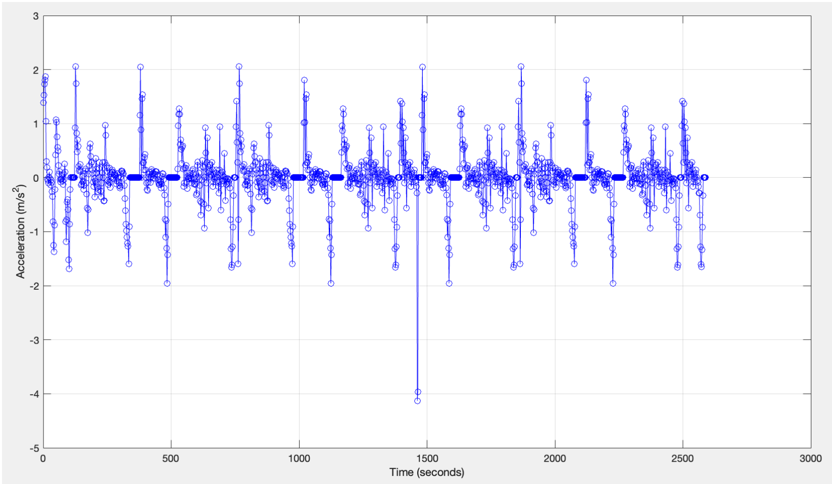

Figure 18. Plot of the Car Acceleration.

f) Use the Max(x) command in Matlab to detect the largest speed during the journey. Find the time when the maximum speed is recorded. Display the maximum speed and the time in the Command window (use the DISP command).

Command Window

```
The max speed is  70.7 km/hr
fx >>
```

```
% Find max. speed and time

[maxSpeed indexMax] = max(Speedkmh);
timeMaxSpeed = indexMax * 2;
clc
disp(['The max speed is  ', num2str(maxSpeed), ' km/hr'])
```

Figure 19. Code to Find the Maximum Speed (km/hr) and the Time.

g) Find the average speed in meters per second of the car for the complete profile. Use the Matlab function MEAN(x) to get the average speed for all values.

```
% Calculate the mean speed

meanSpeed = mean(Speedms);
indicesBelow5mph = find(Speedmph<=5);
countBelow5mph= length(indicesBelow5mph)*2;

disp(['The mean speed is  ', num2str(meanSpeed), ' m/s'])
disp(['The car travels  ', num2str(countBelow5mph), ' seconds below 5
```

Figure 20. Code to Find Mean Speed and the Number of Seconds Below 5 mph.

h) Find the number of seconds the car is traveling below 5 mph. Use the Matlab FIND function to do this part.

```
The mean speed is  12.6887 m/s
The car travels  424 seconds below 5 mph
```