

Assignment 6: Matlab Array Calculations

Date Due: October 18, 2013

Instructor: Trani

Problem 1

Task 1

Create a Matlab script to:

a) Load the Rainfall Rate data file in Matlab. The data contains two columns as shown below.

```
% Rainfall rate data for the month of July in Santo Domingo, Dominican Republic
```

```
%
```

```
% First column is the observation number
```

```
% The second column is the actual rainfall rate in inches
```

```
1.0000 4.3712
```

```
2.0000 5.0026
```

```
3.0000 6.9719
```

```
4.0000 4.8237
```

The data clearly illustrates that rainfall rate is a random variable.

b) Create two distinct variables containing the numerical values of the two columns of the data file loaded in part (a). The first vector should contain all elements of column 1 and should be labeled "observation". The second vector contains the values of rainfall rate and should be labeled "rainfallRate".

Task 2

Enhance the script of Task 1. Using the "hist" command in Matlab, plot a histogram with 20 bins of the rainfall rate data. Label your axes and headings accordingly. Comment on the shape of the rainfall rate data distribution of the data. Is there a central tendency? Comment.

Task 3

Enhance the script of Task 2 to calculate the mean and standard deviation of rainfall rate using the Matlab commands "mean" and "std", respectively. Output the values of the mean and standard deviation to the command window using the Matlab "disp" command. Comment on the results obtained.

Task 4

Enhance the script of Task 3. Create a new variable that saves the rainfall rate data sorted from low to high values. use the Matlab command "sort(x)" where x is the vector that contains the data to be sorted. Make a "stem" plot with sorted rainfall rate in the (y-axis) and the observation in the (x-axis). Add labels as needed.

Task 5

Enhance the script of Task 4. Create a Cumulative Density Function (CDF) plot (read Wikipedia article: http://en.wikipedia.org/wiki/Cumulative_distribution_function). The Matlab command to generate the CDF plot is:

```
cdfplot(x)
```

where x is the random variable. Label the axes correctly: x-axis is the rainfall rate (inches) and y-axis is the cumulative distribution function (F(x)).

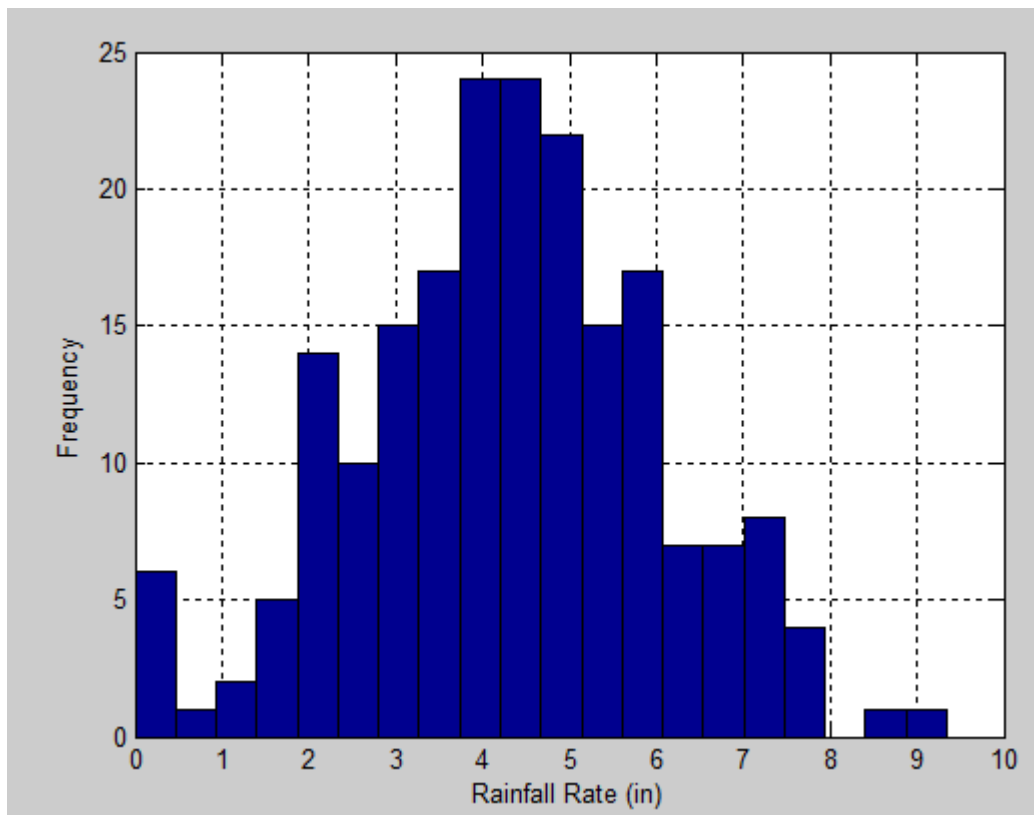
Task 6

Use the index variables analysis presented in class to find the number of rainfall rate observations above 6 inches. After finding the number of observations above 6 inches, display the number in the command window (use the “disp” command).

```
% Script to work with rainfall rate data for the month of July  
% in Santo Domingo, Dominican Republic
```

```
% TASK 1  
% Part a) Load the rainfall data  
load rainfall.m  
  
% Part b) Create two distinct variables  
observation = rainfall(:,1);  
rainfallRate = rainfall(:,2);  
  
% TASK 2: Make a histogram of the rainfall rates  
noBins = 20;
```

```
figure  
hist(rainfallRate,noBins);  
xlabel('Rainfall Rate (in)')  
ylabel('Frequency')  
grid
```



```
% TASK 3: Calculate the mean and standard deviation of the rainfall rates  
meanRainfallRate = mean(rainfallRate);  
X = ['The mean rainfall rate is ', num2str(meanRainfallRate), ' in'];
```

```

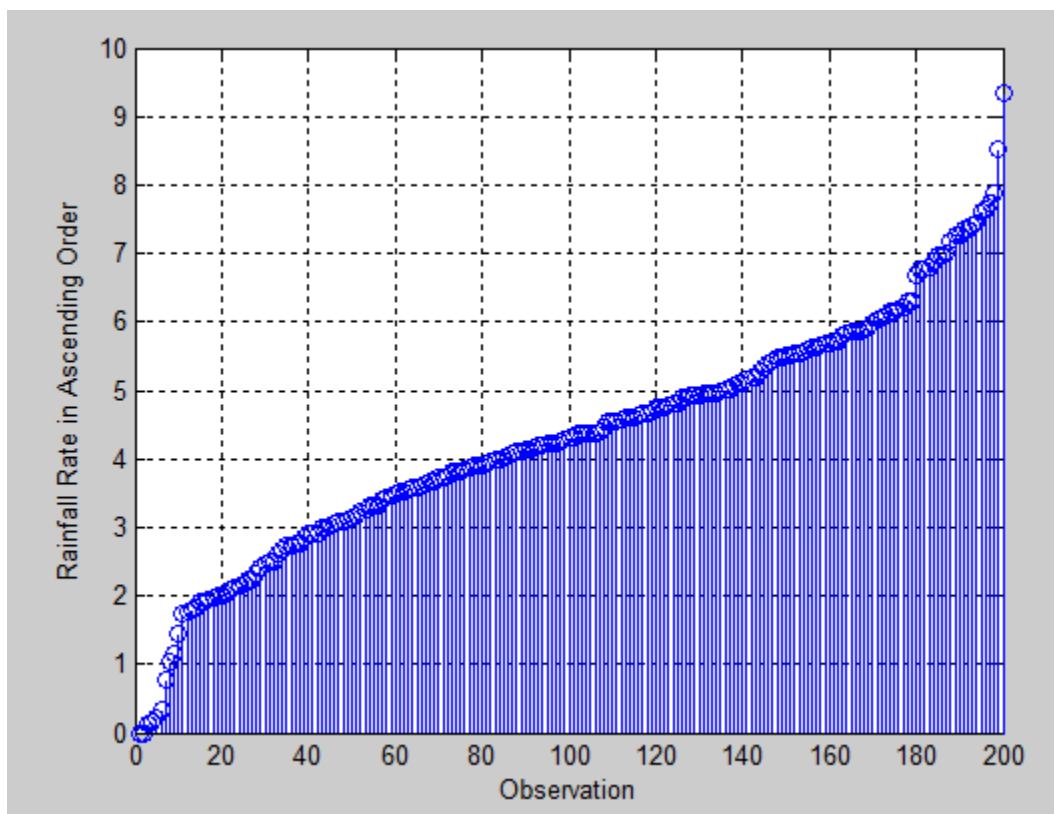
disp(X)

stdevRainfallRate = std(rainfallRate);
Y = ['The standard deviation of the rainfall rate is ',
num2str(stdevRainfallRate), ' in'];
disp(Y)

% TASK 4: Sort the rainfall rates into ascending order and make a plot
rainfallRateAscending = sort(rainfallRate);

figure
stem(observation,rainfallRateAscending);
xlabel('Observation')
ylabel('Rainfall Rate in Ascending Order')
grid

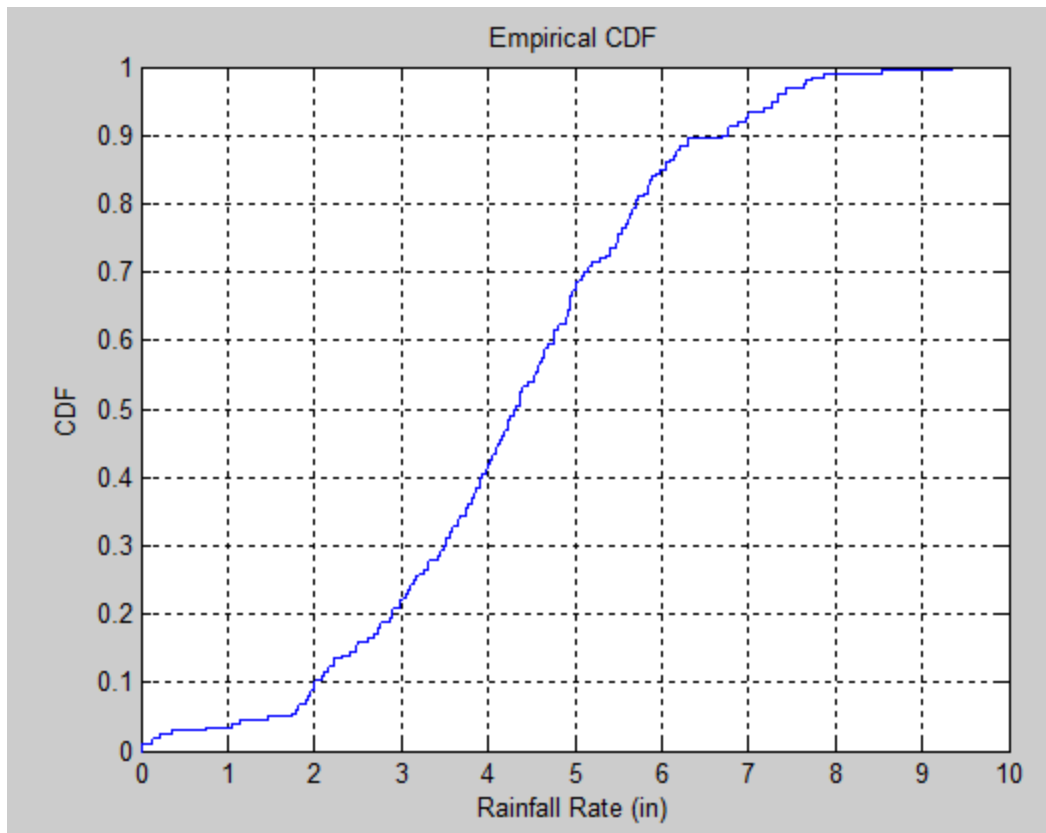
```



```

% TASK 5: Make a CDF of the rainfall rates
figure
cdfplot(rainfallRate)
xlabel('Rainfall Rate (in)')
ylabel('CDF')

```



```
% TASK 6: Find the number of observations of rainfall rates > 6 in
indices = find(rainfallRate>6);
observationsAboveSixInches = length(indices);
Z = ['There are ', num2str(observationsAboveSixInches), ' observations above
6 in.'];
disp(Z)
```

```
Command Window
>> rainfall_script
The mean rainfall rate is 4.3076 in
The standard deviation of the rainfall rate is 1.7428 in
There are 30 observations above 6 in.
fx >>
```

Observe the CDF plot and answer the following:

- a) What is the probability that rainfall rates in the future are less than 3 inches (per month)?

Probability \approx 0.2 from visual inspection of the cdf plot above

- b) If an engineer wants to design a storm water system to handle 95% of the rainfall rates without a ponding (storage) system. Determine the rainfall rate for this design.

Design rainfall rate \approx 7.5 in from visual inspection of the cdf plot above

Problem 2

A formula to estimate the noise generated by rail vehicles is,

$$Leq = SEL_{ref} + 10 \log(N_{cars}) + 20 \log(S/50) + 10 \log(V) - 31.4$$

where:

Leq = equivalent noise level (decibels - dBA)

SEL_{ref} = reference sound exposure level (decibels - dBA)

N_{cars} = number of cars in the train

S = train speed (mph)

V = hourly average train volume (trains per hour)

Task 1:

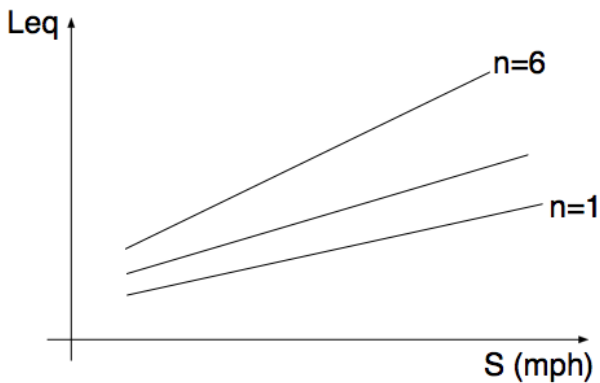
Write a Matlab script to calculate the value of Leq given values of S (speed), SEL_{ref} (sound exposure level), N_{cars} (train cars), and hourly train volume (V). Assume the value of SEL_{ref} to be 76 dBA for this train. Display the result in the command window and write the following statement: " The Value of Equivalent Noise Level is: yyyyy (db)".

Task 2:

Test the Matlab script created in Task 1 to calculate the value of Leq . In your calculations assume the train has 6 cars, the hourly train volume is 30 trains/hr and the train travels at 45 mph.

Task 3:

Modify the Matlab script to create solutions for Leq (in y-axis) as a function of train speed (to be plotted in the x-axis). The idea is to generate solutions for Leq for speeds ranging from 0-70 mph for 1, 2, 3, 4, 5 and 6 cars as shown in the sketch below. Use Matlab vector operations and not a FOR loop.



```

% Function to estimate the noise generated by rail vehicles

% Inputs:
%     SEL ref = reference sound exposure level (decibels - dBA)
%     Ncars = number of cars in the train
%     S = train speed (mph)
%     V = hourly average train volume (trains per hour)

% Outputs:
%     Leq = equivalent noise level (decibels - dBA)

function [Leq] = equivalentNoise(SELref,Ncars,speed,volume)

Leq = SELref + 10.*log(Ncars) + 20.*log(speed/50) + 10.*log(volume) - 31.4;

end

```

```

% Script to demo equivalentNoise.m function

```

```

% TASKS 1 AND 2

```

```

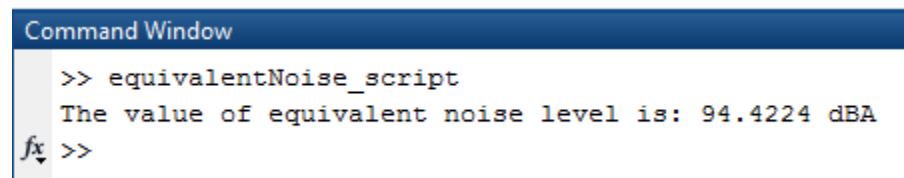
noise = equivalentNoise(76,6,45,30);

```

```

A = ['The value of equivalent noise level is: ', num2str(noise), ' dBA'];
disp(A)

```



```

Command Window
>> equivalentNoise_script
The value of equivalent noise level is: 94.4224 dBA
fx >>

```

```

% TASK 3

```

```

ref = 76;
carsPerTrain = 1:1:6;
trainSpeeds = 0:1:70;
    trainSpeeds = trainSpeeds';
trainsPerHour = 30;

```

```

% Call the equivalentNoise function for each train length

```

```

Leq1 = equivalentNoise(ref,carsPerTrain(1),trainSpeeds,trainsPerHour);
Leq2 = equivalentNoise(ref,carsPerTrain(2),trainSpeeds,trainsPerHour);
Leq3 = equivalentNoise(ref,carsPerTrain(3),trainSpeeds,trainsPerHour);
Leq4 = equivalentNoise(ref,carsPerTrain(4),trainSpeeds,trainsPerHour);
Leq5 = equivalentNoise(ref,carsPerTrain(5),trainSpeeds,trainsPerHour);
Leq6 = equivalentNoise(ref,carsPerTrain(6),trainSpeeds,trainsPerHour);

```

```

% Optional code:

```

```

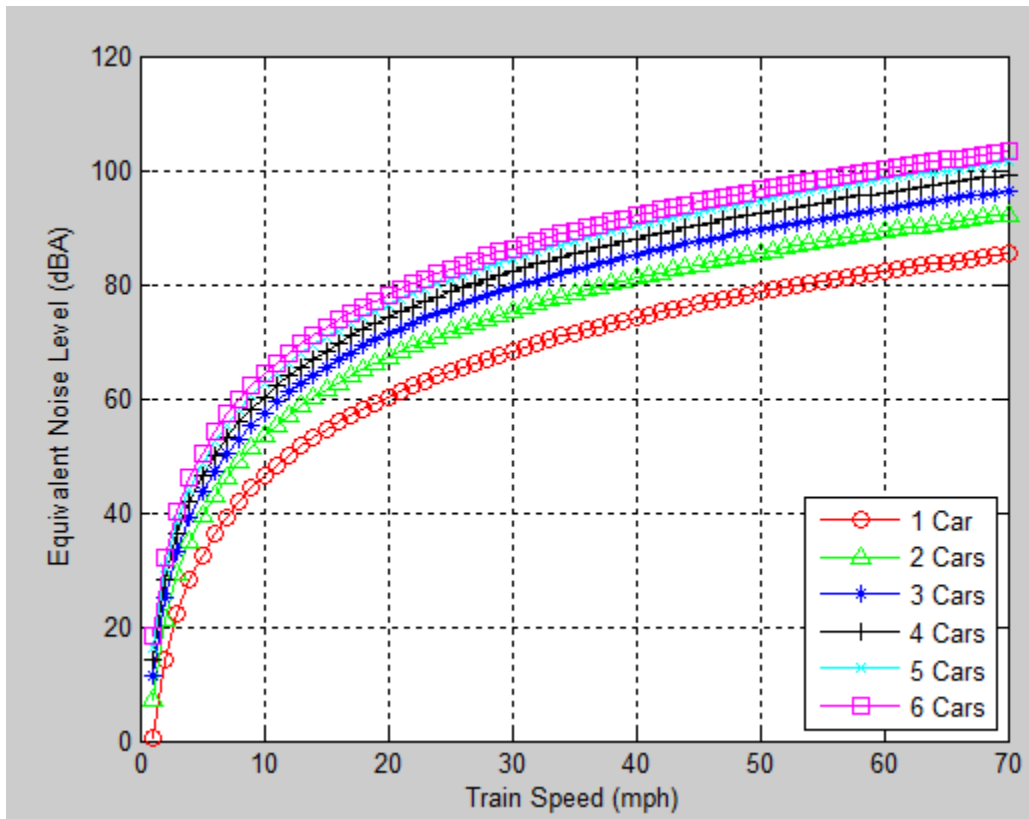
% Leq_matrix = [Leq1 Leq2 Leq3 Leq4 Leq5 Leq6];
% disp(Leq_matrix)

```

```

figure
plot(trainSpeeds,Leq1,'o-r')
grid
xlabel('Train Speed (mph)')
ylabel('Equivalent Noise Level (dBA)')
hold on
plot(trainSpeeds,Leq2,'^-g')
plot(trainSpeeds,Leq3,'*-b')
plot(trainSpeeds,Leq4,'+-k')
plot(trainSpeeds,Leq5,'x-c')
plot(trainSpeeds,Leq6,'s-m')
legend('1 Car','2 Cars','3 Cars','4 Cars','5 Cars','6
Cars','Location','SouthEast')

```



Problem 3

The equation to predict the drag (D) of a high-speed train moving at speed (v) is given by the formula:

$$D = \frac{1}{2} \rho v^2 A C_d$$

where: ρ is the density of air (1.225 kg/cu.meter at sea level conditions), A is the reference area of the high-speed train (in sq. meters) and C_d is a non-dimensional drag coefficient unique to each train. Typically, the value of C_d is derived empirically from wind tunnel tests. In the formula above the units of v are meters per second and the units of D are in Newtons (since drag is a force).

The power required (P) by the train engine to overcome the total resistance is:

$$P = Dv = \frac{1}{2} \rho v^3 A C_d$$

A modern high-speed train has the following characteristics:

$A = 8.4$ square meters

$C_d = 0.26$ (dimensionless)

Task 1:

Create a Matlab script calculates the train drag and power given speed, reference area, density and drag coefficient. This script should estimate the drag force and the power required on the high-speed train for a range of velocities ranging from 1 to 85 m/s at steps of 0.001 m/s. Use Matlab vector operations and not a FOR loop.

In this script make a plot of speed vs. Drag and a second plot of speed vs. Power. Arrange these two plots in a single Figure frame using the subplot command.



```
% Function for Problem 3
function [drag,power] = dragPower(velocity)

% Define global variables
global rho A Cd

drag = 0.5 * rho .* (velocity .^ 2) * A * Cd;

power = drag .* velocity;

end
```



```

% Script for Problem 3

% Define global variables
global rho A Cd

rho = 1.225;    % kg/m^3
A = 8.4;       % m^2
Cd = 0.26;     % dim

% Create the velocity vector
v = 1:0.001:85; % m/s

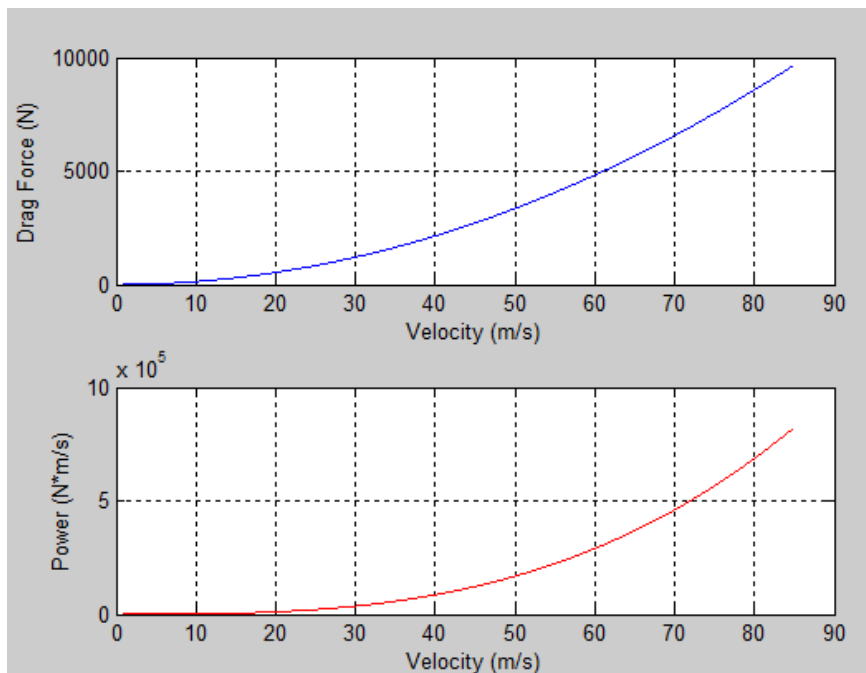
% Call the dragPower function
[D,P] = dragPower(v');

%D = dragPowerMatrix(:,1);
%P = dragPowerMatrix(:,2);

% Plot drag vs. velocity
figure
subplot(2,1,1);
plot(v,D,'-b')
xlabel('Velocity (m/s)')
ylabel('Drag Force (N)')
grid

% Plot power vs velocity
subplot(2,1,2);
plot(v,P,'-r')
xlabel('Velocity (m/s)')
ylabel('Power (N*m/s)')
grid

```



Task 2:

Modify the script created in Task 1 and query the user for the values of Area (A), Density (ρ) and drag coefficient (C_d). Test your script with values of A=7.0, Cd=0.26 and $\rho=1.20$ kg/cu.meter.

```
% Script for Problem 3

% Define global variables
global rho A Cd

rho = input('Please enter the density of air in kg/m^3: ');
A = input('Please enter the reference area of the train in m^2: ');
Cd = input('Please enter the drag coefficient of the train: ');

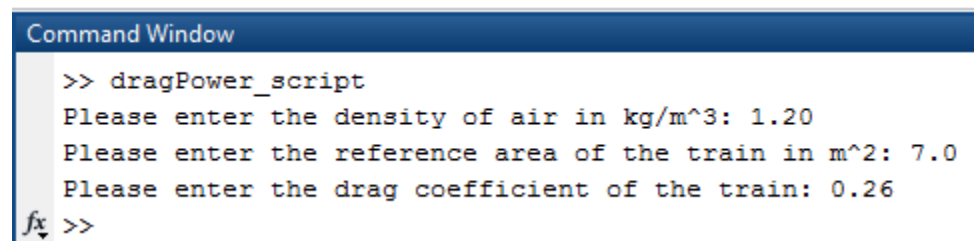
% Create the velocity vector
v = 1:0.001:85; % m/s

% Call the dragPower function
[D,P] = dragPower(v');

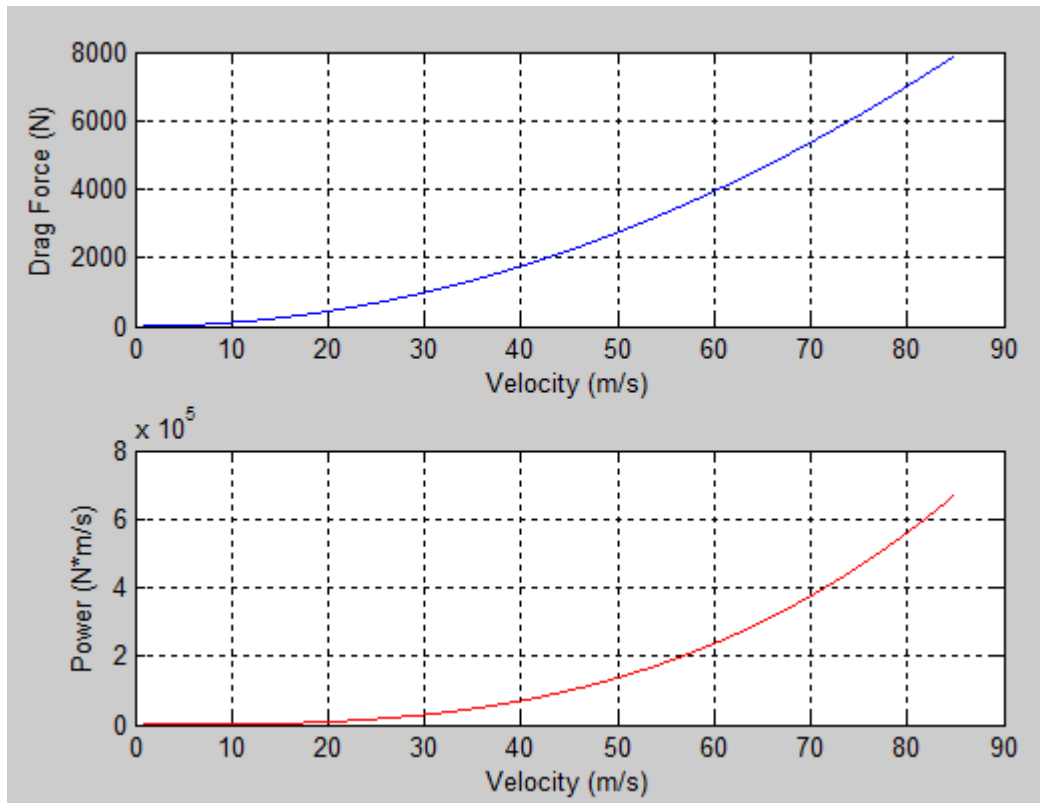
%D = dragPowerMatrix(:,1);
%P = dragPowerMatrix(:,2);

% Plot drag vs. velocity
figure
subplot(2,1,1);
plot(v,D,'-b')
xlabel('Velocity (m/s)')
ylabel('Drag Force (N)')
grid

% Plot power vs velocity
subplot(2,1,2);
plot(v,P,'-r')
xlabel('Velocity (m/s)')
ylabel('Power (N*m/s)')
grid
```



```
Command Window
>> dragPower_script
Please enter the density of air in kg/m^3: 1.20
Please enter the reference area of the train in m^2: 7.0
Please enter the drag coefficient of the train: 0.26
fx >>
```



Problem 4

Repeat the Example Problem # 3 shown in the course notes (**Notes 11 - Matlab Matrices**) using problem partitioning. In other words, create a Matlab script that calculates the moment and deflection equations in two sections: a) a section of the beam to the left of the applied load and b) to the right of the applied load. The calculations should avoid using the FOR LOOP demonstrated in Example # 3.

```
% Function to calculate the deflection and moment along a beam of length L
% that is cantilever on one side, supported on the other side, and subject
% to a point loading W at point a along the beam.
```

```
function [x,deflection,moment] = deflectionMoment(L,a)
```

```
% Define global variables
global E I W R1 R2
```

```
% Choose a horizontal distance increment
increment = 1; % in
```

```
% Establish the horizontal distance vector x, and make sure that the number
% of decimal places of L and a are rounded to the same number of decimal
% places as the increment
```

```
L_rounded = round(L/increment)*increment;
a_rounded = round(a/increment)*increment;
```

```
x_left = 0:increment:a_rounded;
x_right = a_rounded:increment:L_rounded;
```

```

x = [x_left x_right]';

% Calculate beam deflection

% Deflection vector for x < a (Equation from Notes 11, Slide 20c)
deflectionLeft = (1/(6*E*I)) * ((3*R1*(L_rouned^2)*x_left) - ...
    (R1*(x_left.^3)) - (3*W*((L_rouned - a_rouned)^2)*x_left));

% Deflection vector for x > a (Equation from Notes 11, Slide 20c)
deflectionRight = (1/(6*E*I)) * ((R2*(2*L_rouned^3 - ...
    3*(L_rouned^2)*x_right + x_right.^3)) - ...
    (3*W*a*(L_rouned - x_right).^2));

deflection = [deflectionLeft deflectionRight]';

% Calculate moment along the beam

% Moment vector for x < a (Equation from Notes 11, Slide 20e)
momentLeft = W*x_left.*(3*((L_rouned - a_rouned)^2)*L_rouned - ...
    (L_rouned - a_rouned)^3) / (2*L_rouned^3);

% Moment vector for x > a (Equation from Notes 11, Slide 20e)
momentRight = W*x_right.*(3*(L_rouned - a_rouned)^2)*L_rouned - ...
    (L_rouned - a_rouned)^3) / (2*L_rouned^3) - W*(x_right - a_rouned);

moment = [momentLeft momentRight]';

end

% Script to determine deflection and moment profiles for a beam of length L
% that is cantilever on one side, supported on the other side, and subject
% to a point loading W at point a along the beam.

% Begin clocking CPU time
tic;

% Define global variables
global E I W R1 R2

W = 2000; % load in pounds
a = 50; % left distance from single support to load
L = 200; % length of beam
I = 50; % beam moment of inertia (in^4)
E = 29e6; % Young's Modulus (psi)

% Calculate the reaction forces
R1 = W*(3*((L-a)^2)*L - (L-a)^3) / (2*L^3); % reaction at support
R2 = W*(3*a*(L^2) - a^3) / (2*L^3); % reaction at wall

```

```

% call the deflectionMoment function created for this problem
[x,deflection,moment] = deflectionMoment(L,a);

% Plot deflection vs. distance
figure
plot(x,deflection,'o')
xlabel('Distance along the beam x (in)')
ylabel('Deflection (in)')
grid

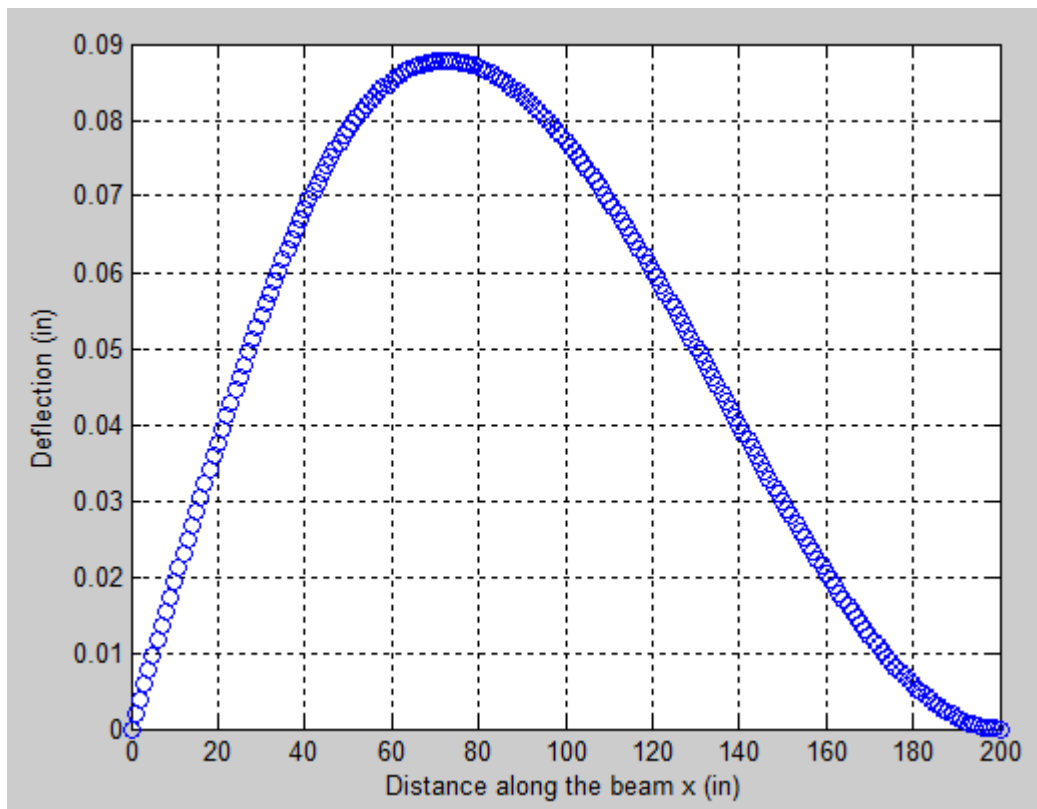
% Plot moment vs. distance
figure
plot(x,moment,'o')
xlabel('Distance along the beam x (in)')
ylabel('Moment (lb-in)')
grid

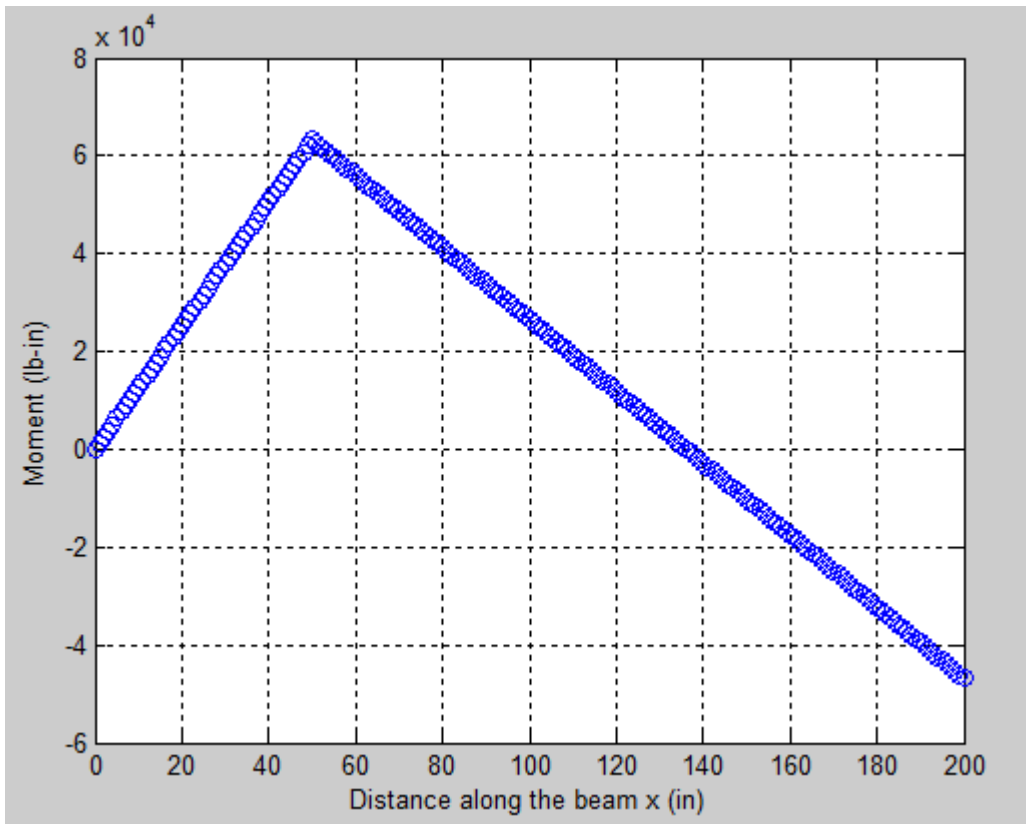
% Stop clocking CPU time and display the CPU time
t = toc;
disp(['CPU time = ', num2str(t), ' s'])

```

Task 1:

Plot the deflection and moment at every section of the beam (every 1 inch).\





Task 2:

Find the CPU time to do the calculations in this new script.

```
Command Window
>> deflectionMoment_script
CPU time = 0.089671 s
fx >> |
```