EGR 103L - Fall 2016

Laboratory 3 - Program Control and Functions

CEMAL YAGCIOGLU (cy111)
Lab 05,Wednesda 11.45 AM - 2.35 PM
25 September, 2016

I understand and have adhered to all the tenets of the Duke Community Standard in completing every part of this assignment. I understand that a violation of any part of the Standard on any part of this assignment can result in failure of this assignment, failure of this course, and/or suspension from Duke University.

Contents

List of Figures

1 Chapra Problem 2.6

The equation for the charge on the capacitor[1, p. 44] is:

$$q(t) = q_0 e^{-Rt/(2L)} \cos\left[\sqrt{\frac{1}{LC} - \left(\frac{R}{2L}\right)^2}\, t\right]$$

The graph of 'Charge on the Capacitor Over Time' has a sinusoidal curve with a decreasing range over time. As the time passes the oscillation of charge on the capacitor decreases, and gets closer to a constant.

2 Based on Chapra Problem 2.22

The equations for the butterfly curve[1, p. 47] are:

$$x = \sin(t)\left(e^{\cos t} - 2\cos 4t - \sin^5 \frac{t}{12}\right)$$
$$y = \cos(t)\left(e^{\cos t} - 2\cos 4t - \sin^5 \frac{t}{12}\right)$$

As they have both x and y have sinusoidal graphes, together they form a equation for 'r', radius from the origin point that changes its direction and magnitude according to the functions. Graph is reflective over y-axis, and r has a larger value while x is positive. Due to these two factors, the curve for y positive values is larger compared to the curve for y negative values, creating a figure to a butterfly.

3 Chapra Problem 3.10

Minimum displacement is at the point (8.7076,-32.741).Thus, it is -32.741 feet. Maximum displacement is at the point (5.7019,195.32).Thus, it is 195.32 feet.

4 Palm Problem 4.44

The a and b values for six gases are, from references [2] and [3] and alphabetically by gas name:

Gas	a(L^2-atm/mol^2)	b(L/mol)
Chlorine,Cl_2	6.49	0.0562
Ammonia,NH_3	4.225	0.03713
Carbon dioxide,CO_2	3.59	0.0427
Oxygen,O_2	1.36	0.0427
Helium,He	0.0341	0.0237
Hydrogen,H_2	0.244	0.0266

A graph of pressures for the above gases at T =300K and specific volumes $\hat{V}$ between 1 and 2 L/mol is presented in Figure 3 on page 13. The pressure for each gas comes from the following function given in Palm [2, p. 215]:

$$P = \frac{RT}{\hat{V} - b} - \frac{a}{\hat{V}^2}$$

5 Palm Problem 4.19

See Table 2 on page 11.

A Codes

A.1 PlotCharge.m

```
% [PlotCharge.m]
% [Cemal Yagcioglu]
% [24 September, 2016]
% Based on: PlotCharge.m
% Written by: Michael R. Gustafson II

% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]


%% Initialize workspace
clear; format short e

%% Make plot
% Initialize plot
figure(1); clf
% Generate time base
t = linspace(0, .8, 1000);
% Calculate charge values using function
q = Charge(t, 10, 60, 9, .000005);
% Make plot
plot(t, q, 'k-')
title('Charge On The Capacitor Over Time(cy111)')
xlabel('Time');
ylabel('Charge');
grid on
print -deps ChargePlot
% Add commands for labels, titles, grid, and saving plot here
```

A.2 Charge.m

```
function qout = Charge(t, q0, R, L, C)
% I understand and have adhered to all the tenets of the Duke
% Community Standard in completing every part of this assignment.  I
% understand that a violation of any part of the Standard on any part
% of this assignment can result in failure of this assignment, failure
% of this course, and/or suspension from Duke University.

%% Error checking

if nargin<5
   error('Not enough inputs!')
elseif isscalar(q0)==0 || isscalar(R)==0 || isscalar(L)==0 || isscalar(C)==0
    error('Non-scalar constants!')
end

%% Calculation


qout = q0.*exp((-R.*t)./(2.*L)).*cos(t.*sqrt((1./(L.*C))-(((R)./(2.*L)).^2)))

```

```

end
```

A.3 RunButterfly.m

```
% [RunButterfly.m]
% [Cemal Yagcioglu]
% [September 24, 2016]
%
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]

%% Initialize workspace
clear, figure(1), clf

%% Set up variable to store times and use Butterfly
t = linspace(0,100,2000);
%b = linspace(0,100,2000);


%  function to get vectors for x and y coordinates
[v, y] = Butterfly(t);
disp(v) %v has the x values
disp(y) %y has the y values

plot(v,y,'k')
title('Butterfly Graph(cy111)')
xlabel('x')
ylabel('y')
grid on


%% Make plot, add grid, labels and titles, then print

print -deps ButterflyPlot % You're welcome!
```

A.4 Butterfly.m

```
function [x, y] = Butterfly(t)
% [Butterfly.m]
% [Cemal Yagcioglu]
% [September 24, 2016]
%
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]

%% Equations for x and for y in terms of t here
x = sin(t).*((exp(cos(t)))-(2.*cos(4.*t))-(sin(t./12).^5));
y = cos(t).*((exp(cos(t)))-(2.*cos(4.*t))-(sin(t./12).^5));
end
```

A.5 TestSingularity.m

```
% [TestSingularity.m]
```

```
% [Cemal Yagcioglu]
% [September 24, 2016]
% Based on: TestSingularity.m
% Written by: Michael R. Gustafson II
%
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]

%% Change the code in lines 14 and 21

clear; format short e
%Singularity = @(x, a, n) (-5/6).*(((x-0).^4)-((x-5).^4))+15/6.*(x-8).^3+75.*(x-7).^2+57/6.*x.^3-238
%Singularity = @(x, a, n) (x-a).^n;

%MaskForA1 = (a==-1 & n==0);
%FunctionForA1 = 0;

%Singularity = @(x, a, n) ((x-a).^n).*((x>=-1 & a==-1 & n==0)|(x>0 & a==0 & n==1)|(x>1 & a==1 & n==1
Singularity = @(x, a, n) (((x-a).^n).*(x>a))

% Your Code Here!
x  = linspace(-2, 5, 500);



plot(x, Singularity(x, -1, 0), 'k-',...
     x, Singularity(x,  0, 1), 'k--',...
     x, Singularity(x,  1, 1), 'k-.',...
     x, Singularity(x,  3, 2), 'k:');
legend('<x+1>^0', '<x>^1', '<x-1>^1', '<x-3>^2', 'Location','Northwest') % I have the newer version,
                                                                          % to change the given struc
title('Four Different Values of y=<x-a>^n (cy111)'); % Your NetID
xlabel('x');
ylabel('y');
grid off
print -deps SingPlots
```

A.6 BeamDisplacement.m

```
% [Function or Script Name]
% [Cemal Yagcioglu]
% [22 september, 2016]
%
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]

% Honor Code

%% Initialize the workspace

clear; format short e

%% Genreate values and plot them
%  Generate 100 sample points for x, define Singularity, then
```

```
%  calculate the displacement based on those 100 points

x  = linspace(0, 10, 100);
S = @(x, a, n) (x>a).*((x-a).^n); %%I shortened Singularity function to S
Deflection = ((-5/6).*(S(x,0,4)-S(x,5,4))+(5/2).*(S(x,8,3))+(325/2).*(S(x,7,2))+(79/12)...
    .*(S(x,0,3))-(76/3).*(S(x,0,1)));

%  Initialize the plot, plot the values, and add
figure(1); clf
plot(x,Deflection)
%  labels, a title, and a grid.  Print the plot
grid on
title('Deflection u of a Beam (cy111)'); % Your NetID
xlabel('Distance (feet)');
ylabel('Deflection');

print -deps DisplacementPlot

%% Generate more precise values for min/max determination
%  Generate 1e6 sample points for x, then
%  calculate the displacement based on those 1e6 points
%  Note - Singularity was already defined above, just use it

x  = linspace(0, 10, 10^6);
S = @(x, a, n) (x>a).*((x-a).^n); %%I shortened Singularity function to S
Deflection = ((-5/6).*(S(x,0,4)-S(x,5,4))+(5/2).*(S(x,8,3))+(325/2).*(S(x,7,2))+(79/12)...
    .*(S(x,0,3))-(76/3).*(S(x,0,1)));

% Determine most positive and negative displacements and location
x(find(Deflection==max(Deflection)))
max(Deflection)
x(find(Deflection==min(Deflection)))
min(Deflection)
hold on
plot(x(find(Deflection==max(Deflection))),max(Deflection),'k-^')
plot(x(find(Deflection==min(Deflection))),min(Deflection),'k-^')



```

A.7 GraphPressures.m

```
% [GraphPressures.m]
% [Cemal Yagcioglu]
% [September 24, 2016]
%
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]


%% Initialize the workspace
clear; format short e

%% Create vector of volumes and vectors for each gas

```

```
Vol= linspace(1.0,2.0,200);
Temp=300


%% Start and clear figure
figure(1), clf
%% Set increment for point plots and plot points and lines
MyP1 = VanDerWaals(Temp,Vol,'Helium')
plot(Vol,MyP1)
%hold on
MyP2 = VanDerWaals(Temp,Vol,'Hydrogen')
%plot(Vol,MyP2)
MyP3 = VanDerWaals(Temp,Vol,'Oxygen')
MyP4 = VanDerWaals(Temp,Vol,'Chlorine')
%plot(Vol,MyP3) 'Oxygen','Chlorine','Carbon dioxide',
%plot(Vol,MyP4)
MyP5 = VanDerWaals(Temp,Vol,'Carbon dioxide')
%plot(Vol,MyP5)
MyP6 = VanDerWaals(Temp,Vol,'Ammonia')
plot(Vol,MyP1,'k-',Vol,MyP2,'k-',Vol,MyP3,'k-',Vol,MyP4,'k-',Vol,MyP5,'k-',Vol,MyP6,'k-')
Incr = 5;
Indices = 1:Incr:length(Vol);
hold on
PointPlot=plot(...
    Vol(Indices), MyP1(Indices), 'k^',...
    Vol(Indices), MyP2(Indices), 'ko',...
    Vol(Indices), MyP3(Indices), 'kx',...
    Vol(Indices), MyP4(Indices), 'k+',...
    Vol(Indices), MyP5(Indices), 'k*',...
    Vol(Indices), MyP6(Indices), 'kp');
hold off
legend(PointPlot, 'Helium', 'Hydrogen', 'Oxygen' , 'Chlorine' , 'Carbon dioxide', 'Ammonia','Locatio

%% Add title, labels, then print
title('Change in Pressure with Volume(cy111)')
xlabel('Volume (L)')
ylabel('Pressure (atm)')
print -deps PressuresPlot
```

A.8 VanDerWaals.m

```
function Pressure = VanDerWaals(Temp, Vol, Gas)
% VanDerWaals  Calculate pressures for a gas.
%   Pressure = VanDerWaals(Temp, Vol, Gas)
%     Temp: a matrix of temperatures
%     Vol: a matrix of specific volumes
%     Gas: a string with the name of a gas
% [VanDerWaals.m]
% [Cemal Yagcioglu]
% [September 24, 2016]

% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]

%% Use switch tree to determine gas and a and b values
```

```
R = 0.08206;

        switch Gas
                case 'Helium'
                a=0.0341;
                b=0.0237;
                case 'He'
                a=0.0341;
                b=0.0237;
                 case 'Hydrogen'
                a=0.244;
                b=0.0266;
                 case 'H2'
                a=0.244;
                b=0.0266;
                 case 'Oxygen'
                a=1.36;
                b=0.0318;
                 case 'O2'
                a=1.36;
                b=0.0318;
                 case 'Chlorine'
                a=6.49;
                b=0.0562;
                 case 'Cl2'
                a=6.49;
                b=0.0562;
                 case 'Carbon dioxide'
                a=3.59;
                b=0.0427;
                 case 'CO2'
                a=3.59;
                b=0.0427;
                 case 'Ammonia'
                a=4.225;
                b=0.03713;

                 case 'NH3'
                 a=4.225;
                b=0.03713;

                otherwise
        error('Gas not in database!');


        end
        Pressure= (R.*Temp)./(Vol-b)-a./((Vol).^2);
end


%% Use formula to calculate array of pressures for that gas
```

A.9 LeapYears.m

```
function [LeapTimes, GivenYears] = LeapYears(Year1, Year2)

```

```
%Years = [Year1:Year2]

% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]


% Error checking

% Calculations

if(nargin<1)
   error('At least one input is required.')
end

if (nargin==1)
    if(floor(Year1)~=Year1 | length(Year1)~=1)
%&& floor(Year1)~=Year1)
    error('Single integer values only.')
    else
        GivenYears=[Year1];
        LeapTimes=[0];
    end
end
        %%(nargin==1 && ismatrix(Year1)==0 && floor(Year1)==Year1) this
        %%did
        %%not work
    if nargin==2
        if(length(Year2)~=1 || length(Year1)~=1 ||floor(Year2)~=Year2 || floor(Year1)~=Year1)
            error('Single integer values only.')
        end
        if(Year2<Year1)
            error('Invalid range')
        end
        GivenYears=(Year1:Year2);
        LeapTimes=zeros(size(GivenYears));
    end

    for a=1:length(GivenYears)
        if mod(GivenYears(a),400)==0
            LeapTimes(a)=1;
        elseif mod(GivenYears(a),100)~=0 && mod(GivenYears(a),4)==0
            LeapTimes(a)=1;
        end
    end
end
% Error checking

% Calculations

```

B Diary

```
Case 1 error: Not enough inputs!
Case 2 error: Not enough inputs!
Case 3 error: Non-scalar constants!
Case 4 error: Non-scalar constants!
Case 5 error: Non-scalar constants!
Case 6 error: Non-scalar constants!
```

Table 1: Output from TestCharge.m

```
Trial 1:
ERROR: At least one input is required.

Trial 2:
ERROR: Single integer values only.

Trial 3:
ERROR: Single integer values only.

Trial 4:
ERROR: Single integer values only.

Trial 5:
ExtDay: 0
Years: 1800

Trial 6:
ExtDay: 1
Years: 2000

Trial 7:
ERROR: Single integer values only.

Trial 8:
ERROR: Single integer values only.

Trial 9:
ERROR: Single integer values only.

Trial 10:
ERROR: Single integer values only.

Trial 11:
ERROR: Single integer values only.

Trial 12:
ERROR: Single integer values only.

Trial 13:
ERROR: Invalid range

Trial 14:
ExtDay: 0 0 1 0 0 0 0 0 0
Years: 1894 1895 1896 1897 1898 1899 1900 1901 1902

Trial 15:
ExtDay: 0 0 1 0 0 0 1 0 0
Years: 1994 1995 1996 1997 1998 1999 2000 2001 2002

```

Table 2: Output from TestLeapYear.m

C Figures

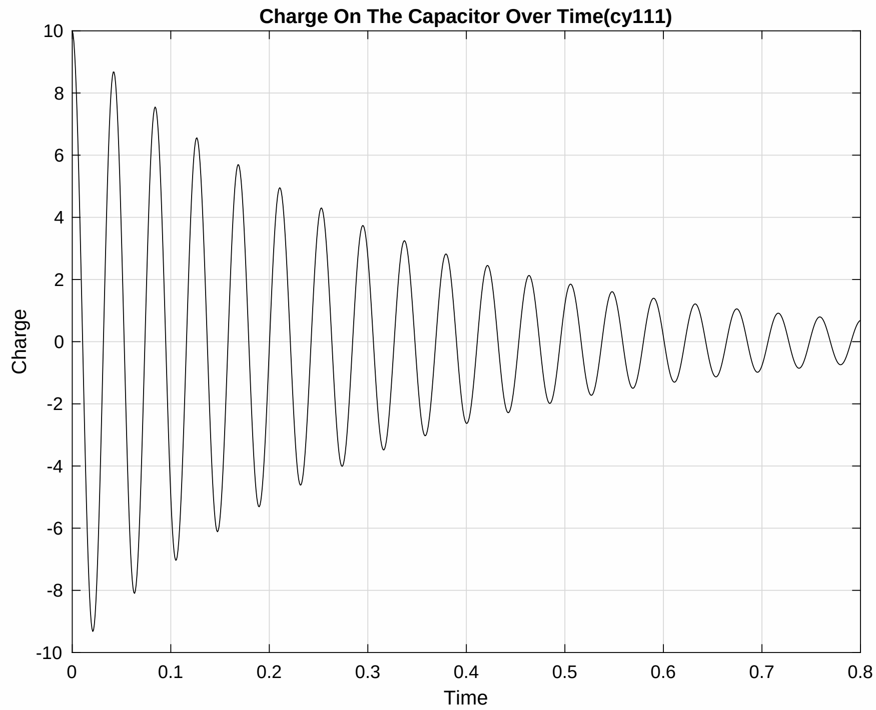

Figure 1: Plot for Chapra 2.9.

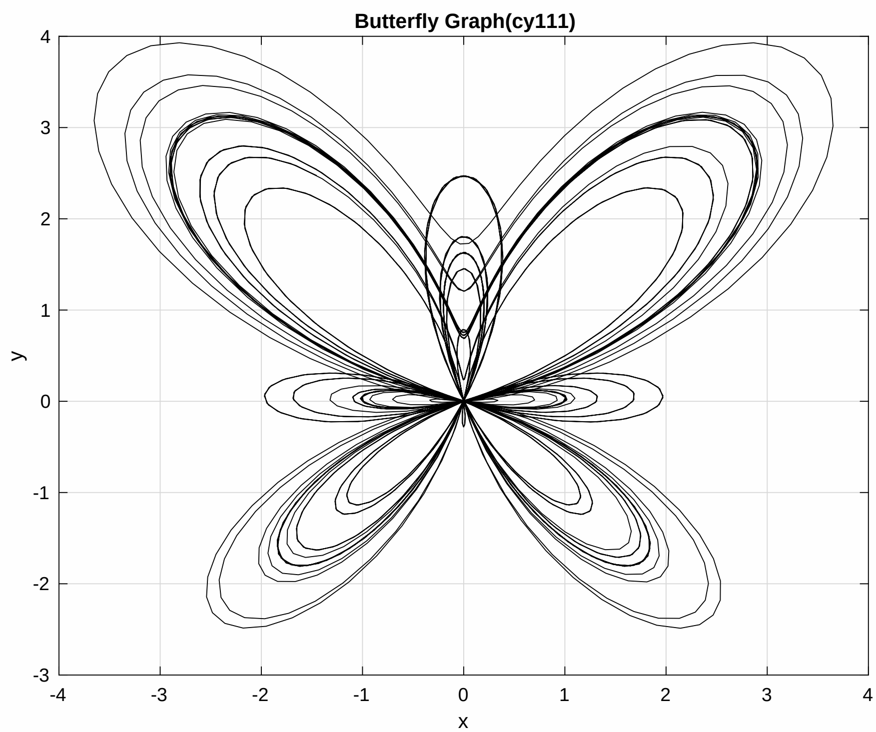

Figure 2: Butterfly Curve.

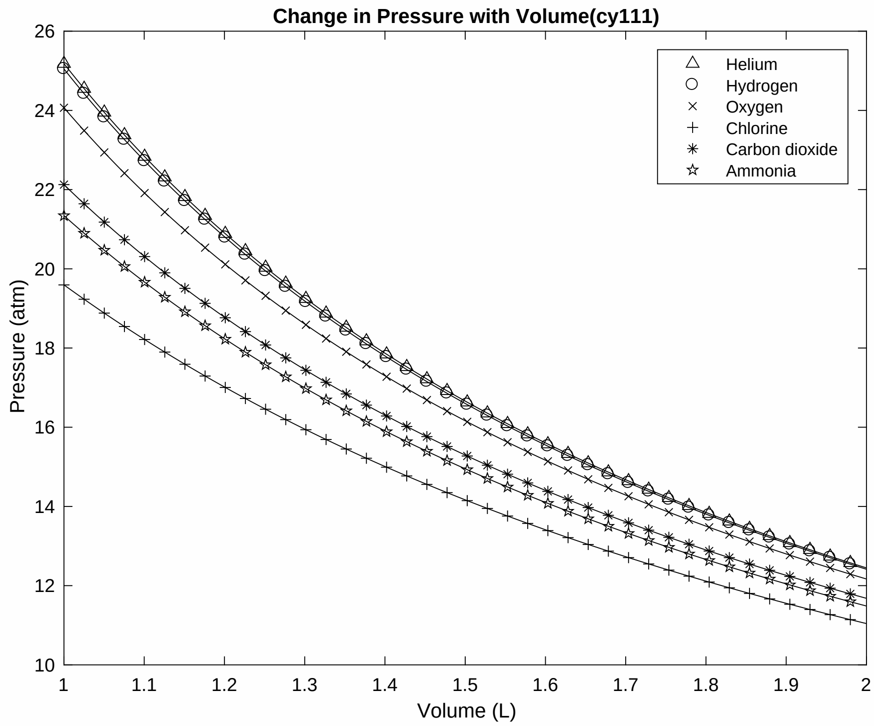

Figure 3: VanDerWaals Calculations Plots.

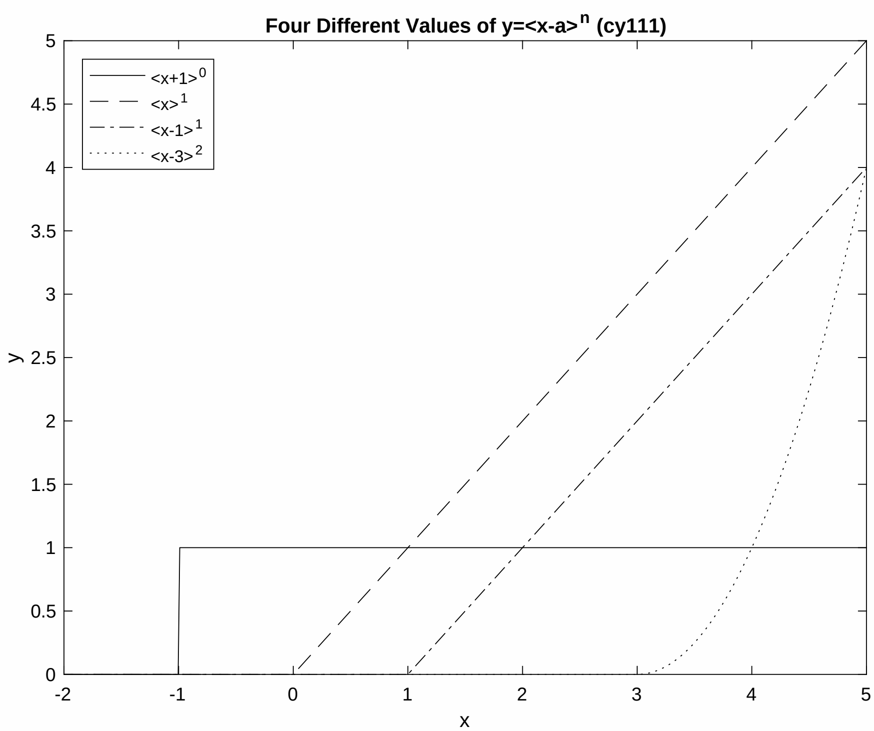

Figure 4: Test of Singularity function.

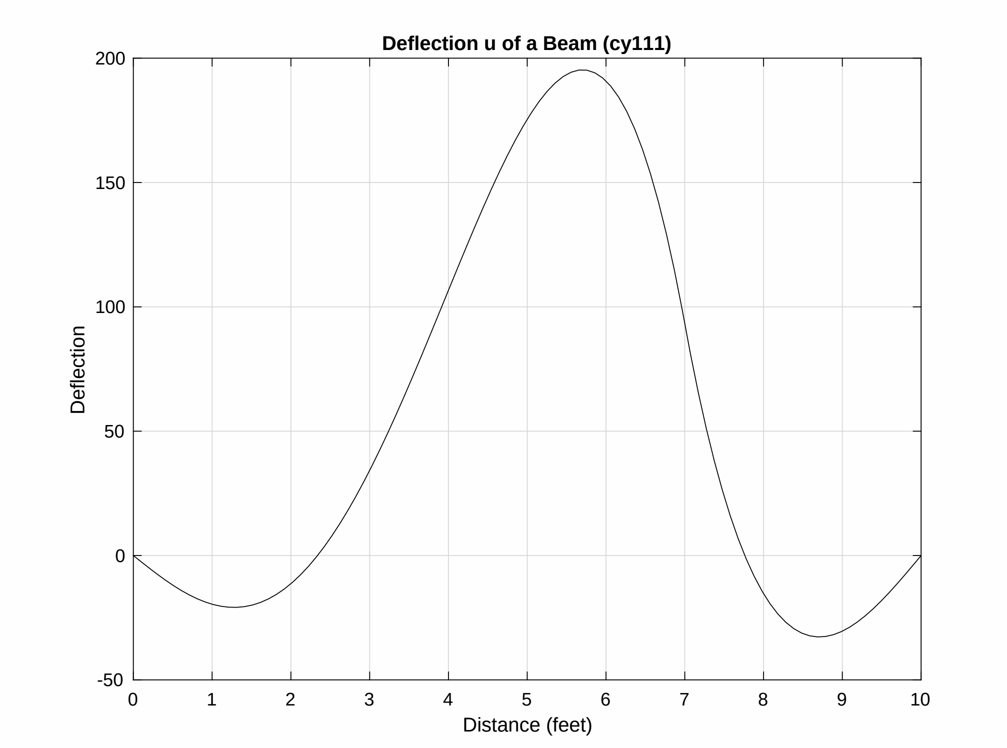

Figure 5: Displacement plot for a beam.

References

[1] Chapra, Steven C., *Applied Numerical Methods with MATLAB for Engineering and Scientists.* McGraw-Hill, New York, 3rd Edition, 2012.

[2] Palm, William J., *Introduction to MATLAB for Engineers.* McGraw-Hill, New York, 3rd Edition, 2011.

[3] *Van Der Waal's Constants for Real Gases.* Chemistry LibreTexts. UC Davis, California, 10 Aug. 2016. Web. 25 Sept. 2016.