ME 594 – Numerical Methods – HW07

Viral Panchal | Due Date: 11/09

"I pledge my honor that I have abided by the Stevens Honor System"

1*. Textbook problem 8.27: The following data is obtained for the velocity of a vehicle during a crash test:

t (ms)	0	10	20	30	40	50	60	70	80
v (mph)	30	29.5	28	23	10	5	2	0.5	0

If the vehicle weight is 2,400 lb, determine the instantaneous force F acting on the vehicle during the crash as a function of time. The force can be calculated by $F = m\frac{dv}{dt}$, and the mass of the car m is 2400/32.2 slug. Note that 1 ms $= 10^{-3}$ s and 1 mile $= 5,280$ ft.

Solve by writing a user-defined function that uses equal Δt's. Do not use any MATLAB intrinsic differentiation functions.

ans. `F=[0.0000e+00 -1.0932e+04 -3.5528e+04 -9.8385e+04 -9.8385e+04 -4.3727e+04 -2.4596e+04 -1.0932e+04 0.0000e+00]`

MATLAB program:

- ***Script for Force.m***

```
% Function to get the force
function F = Force(t,v)

n = length(t);
dt = t(2) - t(1);
W = 2400;
g = 32.2;
m = W/g;
ms_per_s = 1000;
ft_per_mi = 5280;
s_per_h = 3600;

conv = ms_per_s*ft_per_mi/s_per_h;

F(1) = m*(-3*v(1)+4*v(2)-v(3))/(2*dt)*conv;
F(n) = m*(v(n-2)-4*v(n-1)+3*v(n))/(2*dt)*conv;
for i = 2:n-1
   F(i) = m*(v(i+1)-v(i-1))/(2*dt)*conv;
end
```

- ***Driver to run the above function***

```
% Q1 Driver

clear all
close all
clc

t = [0 10 20 30 40 50 60 70 80];
v = [30 29.5 28 23 10 5 2 0.5 0];

F = Force(t,v);
F = F';
disp(F)
```

MATLAB output:

```
   1.0e+04 *

         0
   -1.0932
   -3.5528
   -9.8385
   -9.8385
   -4.3727
   -2.4596
   -1.0932
         0
```

Published with MATLAB® R2021a

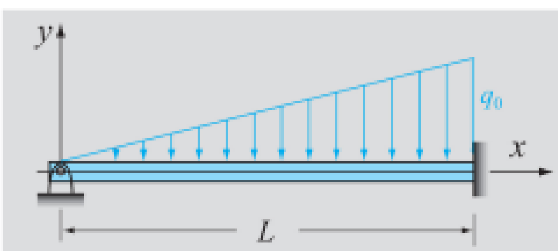
2*. Textbook problem 8.36: A 30 ft long uniform beam is simply supported at the left end and clamped at the right end. The beam is subjected to the triangular load shown. The deflection of the beam is given by the differential equation:

$$\frac{\mathrm{d}^2 y}{\mathrm{d}x^2} = \frac{M(x)}{EI},$$

where y is the deflection, x is the coordinate measured along the length of the beam, $M(x)$ is the bending moment, $E = 29 \times 10^6$ psi is the elastic modulus, and $I = 720\,\text{in}^4$ is its moment of inertia. The following data is obtained from measuring the deflection of the beam versus position:

x (in)	0	24	48	72	96	120	144	168
y (in)	0	-0.111	-0.216	-0.309	-0.386	-0.441	-0.473	-0.479
x (in)	192	216	240	264	288	312	336	360
y (in)	-0.458	-0.412	-0.345	-0.263	-0.174	-0.090	-0.026	0

Using the data, determine the bending moment $M(x)$ in ft · lbf at each location x. Solve the problem by writing a user-defined function that uses a second order scheme at each node. You are not allowed to use a MATLAB intrinsic differentiation function. Make a plot of the bending moment diagram.



ans. `M=[1.6769e-10 1.8125e+04 3.6250e+04 4.8333e+04 6.6458e+04 6.9479e+04 7.8542e+04 8.1562e+04 7.5521e+04 6.3437e+04 4.5312e+04 2.1146e+04 -1.5104e+04 -6.0417e+04 -1.1479e+05 -1.6917e+05]`

MATLAB program:

- ***Function to compute the bending moment***

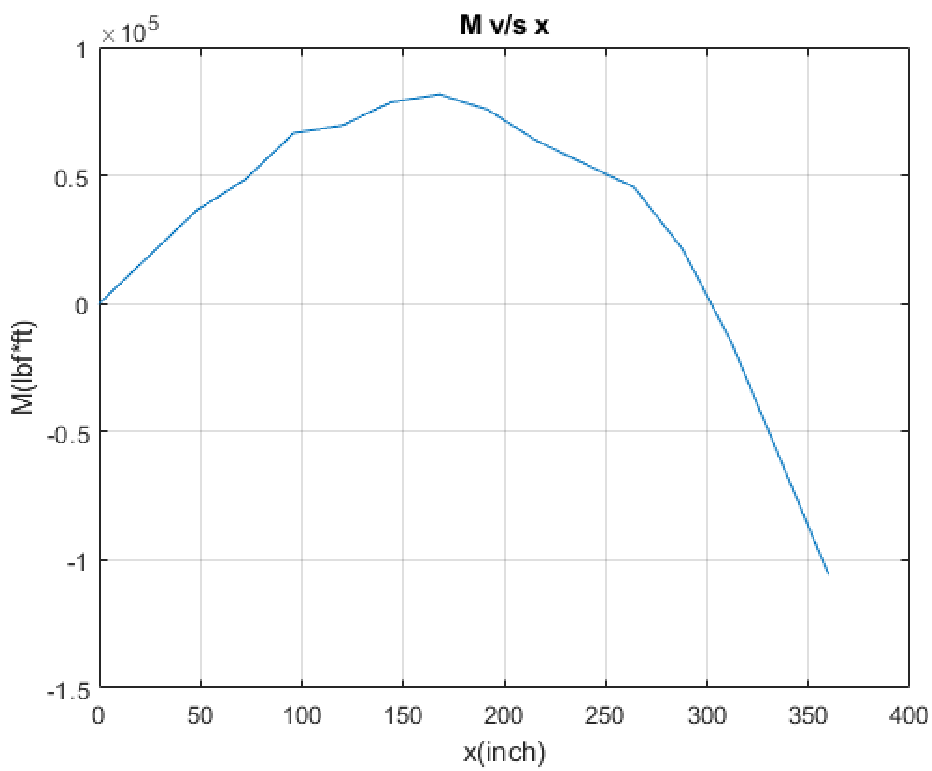
```
% Function to compute the bending moment

function M = BendingMoment(x,y)

n = length(x);
dx = x(2)-x(1);

E = 29*10^6;
I = 720;
I_ft = 12;

M(1)=(2*y(1)-5*y(2)+4*y(3)-y(4))/dx^2*E*I/I_ft;
M(n) =(-1*y(n-3)+4*y(n-2)-5*y(n-1)+2*y(n))/dx^2*E*I/I_ft;

for i = 2:n-1
    M(i) = (y(i-1)-2*y(i)+y(i+1))/dx^2*E*I/I_ft;
end

plot(x,M)
xlabel('x(inch)')
ylabel('M(lbf*ft)')
```

- ***Driver to run the above function:***

```
% Q2 driver

clear all
close all
clc

x = [0 24 48 72 96 120 144 168 192 216 264 288 312 336 360];
y = [0 -0.111 -0.216 -0.309 -.386 -.441 -.473 -.479 -.458 -.412 -.345 -.263 -
.174 -.090 -.026 0];

M = BendingMoment(x,y);
M = M';
disp(M)
```

MATLAB Output:

```
  1.0e+05 *

    0.0000
    0.1813
    0.3625
    0.4833
    0.6646
    0.6948
    0.7854
    0.8156
    0.7552
    0.6344
    0.4531
    0.2115
   -0.1510
   -0.6042
   -1.0573
```



Published with MATLAB® R2021a

3*. Textbook problem 9.26, but slightly modified**: The error function $erf(x)$ (also called the Gauss error function), which is used in various disciplines (e.g., statistics, material science), is defined as:

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt.$$

Write a user-defined MATLAB function that calculates the error function. For function name and arguments use `ef=ErrorFun(x)`. Use the composite Simpson's 1/3 method to evaluate the integral inside `ErrorFun`.

(a) Use `ErrorFun` to make a plot of the error function for $0 \le x \le 2$. The spacing between points on the plot should be 0.02.
(b) Skip this.

MATLAB program:

- ***Script for computing the error and plotting:***

```
% Function to get the error and plot it
function ef = ErrorFun(x)

dt = 0.2;
h = dt/2;
N = x/dt+1;

ef = zeros(1,N);
t = linspace(0,x,N);

ef(1) = 0;
for i = 2:N
    local_simp = h/3*(feval('IntFun',t(i-1))+
4*feval('IntFun',t(i-1)+h) + feval('IntFun',t(i-1)+2*h));
    ef(i) = ef(i-1) + local_simp;
end

ef = ef.*2/sqrt(pi);
plot(t,ef)
title('error v/s x')
xlabel('x')
ylabel('erf(x)')
grid on
```

- ***Integral function for simpson's method***

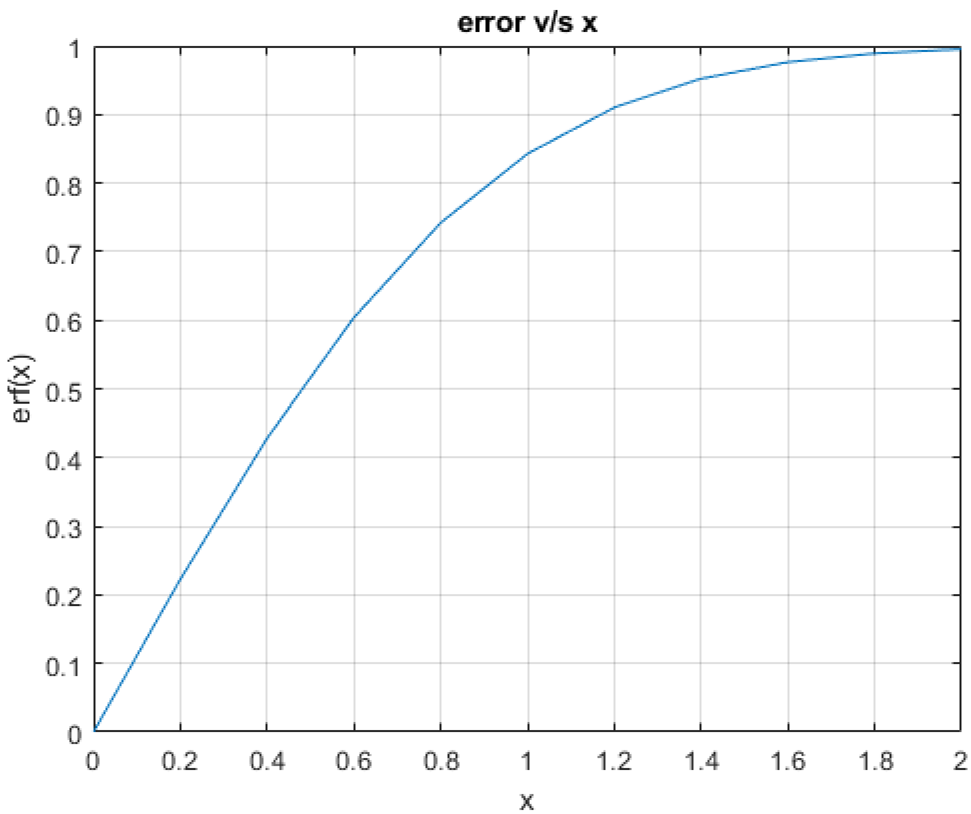
```
% script for intergral function
function f = IntFun(t)

f = exp(-1*t^2);
```

- ***Driver for Q3***

```
% Driver for Q3

clear all
close all
clc

x = 2;
ef = ErrorFun(x);
```

MATLAB Output

```
% Driver for Q3

clear all
close all
clc

x = 2;
ef = ErrorFun(x);
```



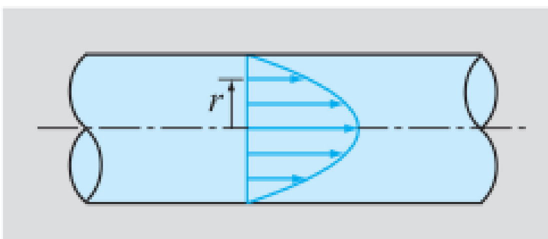
Published with MATLAB® R2021a

4*. Textbook problem 9.32: Measurements of the velocity distribution of a fluid flowing in a pipe (laminar flow) are given in the table. The flow rate Q (volume of fluid per second) in the pipe can be calculated by:

$$Q = \int_0^R 2\pi v r \, \mathrm{d}r,$$

where R is the radius of the pipe. Use the data in the table to evaluate Q.
(a) Use the composite trapezoid method.
(b) Use the composite Simpson's 1/3 method.
(c) Skip this.

ans. `Qt=245.87`, `Qs=249.81`



r (in)	0.0	0.25	0.5	0.75	1	1.25	1.5	1.75	2.0
v (in/s)	38.0	37.6	36.2	33.6	29.7	24.5	17.8	9.6	0

MATLAB program:

- ***Trapezoid function script***

```
% Script to solve using trapezoid method
function Q = Trapezoid(v)

R = 2;
n = length(v);
h = R/(n-1);
r = linspace(0,R,n);

Q = 0;
for i = 1:n-1
    Q = Q+0.5*(v(i)*r(i)+v(i+1)*r(i+1));
end
Q = Q*2*pi*h;
```

- ***Simpson's function script***

```
% Script to solve using Simpson's method
function Q = Simpsons(v)

R = 2;
n = length(v);
h = R/(n-1);
r = linspace(0,R,n);

Q = 0;
for i=1:2:n-2
    Q = Q + (v(i)*r(i)+4*v(i+1)*r(i+1)+v(i+2)*r(i+2));
end
Q = Q*2*pi*h/3;
```

- ***Driver for Q4***

```
% Driver Q4
clear all
close all
clc

v = [38 37.6 36.2 33.6 29.7 24.5 17.8 9.6 0];

Q_t = Trapezoid(v);
Q_s = Simpsons(v);

fprintf('Q_t =')
disp(Q_t);
fprintf('Q_s =')
disp(Q_s);
```

MATLAB Output:

```
Q_t =  245.8689

Q_s =  249.8090
```

Published with MATLAB® R2021a