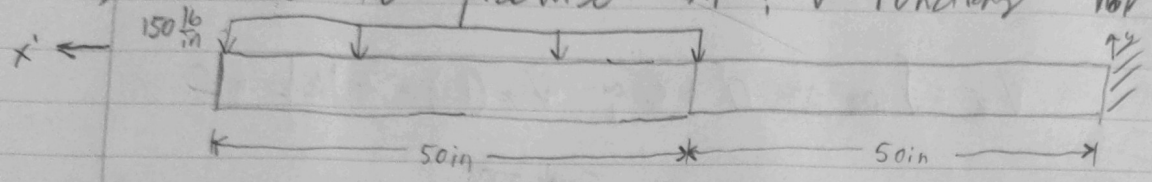
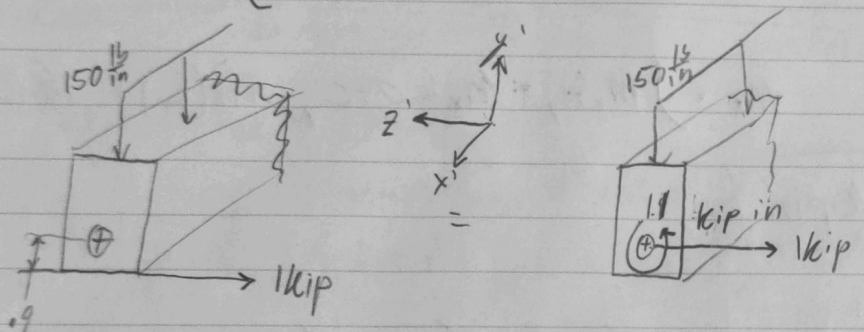
# HW5

a & b) are in Matlab, I wrote a function which computes these values.

c) Solve the piecewise M & V functions for




$$p_y = \begin{cases} -150\ \frac{lb}{in}, & x \geq 50 \\ 0, & x < 50 \end{cases}, \quad p_z = 0, \quad p_x = 0$$




$$m_y = 0, \quad m_x = 0, \quad m_z = 0$$

d) $P_x(L) = 0$, $V_y(L) = 0$, $V_z(L) = -1\ kip$, $M_x(L) = 1.1\ kip\,in$, $M_y(L) = 0$, $M_z(L) = 0$

$P_x' = -p_x$, $P_x = 0 + C$, $P_x(L) = 0 = C$, $\boxed{P_x = 0}$

$V_y' = -p_y$, $V_y = \int \begin{cases} 150\ \frac{lb}{in}, & x \geq 50 \\ 0, & x < 50 \end{cases} = \begin{cases} C_1 + 150\ \frac{lb}{in} x, & x \geq 50 \\ C_2, & x < 50 \end{cases}$

$V_y(L) = 0 = C_1 + 150\,L = C_1 = -150\,L$, $V_y(\frac{L}{2}) = 150\ \frac{lb}{in}(x - L) = -150\ \frac{lb}{in}\ \frac{L}{2}$

$V_y(\frac{L}{2}) = -150\ \frac{lb}{in}\ \frac{L}{2} = C_2$, $C_2 = -150\ \frac{lb}{in}\ \frac{L}{2}$

HW5

$$V_y(x) = \begin{cases} 150(x-L), & x \geq 50 \\ -75L, & x < 50 \end{cases}$$

$V_z' = p_z$, $V_z = \int p_z = 0 + C$, $V_z(L) = -1\,kip = C$

$$V_z(x) = -1\,kip$$

$M_x' = m_x = 0$, $M_x = \int m_x$, $M_x = 0 + C$, $M_x(L) = .9\,kip\,in = C$

$$M_x(x) = 1.9\,kip\,in$$

$M_y' = -m_y + V_z$, $M_y = -\int m_y + \int V_z = -\int 1\,kip = -x\,kip + C$

$M_y(L) = 0 = -L\,kip + C$, $C = +L\,kip$

$$M_y = (L-x)\,kip$$

$$M_z' = -m_z - V_y = \begin{cases} 150(L-x), & x \geq 50 \\ 75L, & x < 50 \end{cases}$$

$$M_z = \int \begin{cases} 150(L-x), & x \geq 50 \\ 75L, & x < 50 \end{cases} = \begin{cases} 150Lx - 75x^2 + C_1, & x \geq 50 \\ 75Lx + C_2, & x < 50 \end{cases}$$

$M_z(L) = 0 = 150L^2 - 75L^2 + C_1 = 75L^2 + C_1$, $C_1 = -75L^2$

# HW 5

$$M_z\left(\frac{L}{2}\right) = 75L^2 - 75\frac{L^2}{4} - 75L^2 = \frac{75}{2}L^2 + C_2$$

$$-75\left(\frac{L^2}{4} + \frac{L^2}{2}\right) = C_2 = -\frac{3}{4}75L^2 = -\frac{225}{4}L^2$$

$$M_z = \begin{cases} 150Lx - 75x^2 - 75L^2, & x \geq 50 \\ 75Lx - \frac{225}{4}L^2, & x < 50 \end{cases}$$

e) $$\varepsilon_{xx} = \frac{P + P^T}{E_1 A^*} - \frac{M_z - M_z^T}{E_1 I_{zz}^*} y + \frac{M_y + M_y^T}{E_1 I_{yy}^*} z$$

$$\sigma_{xx} = E\varepsilon_{xx} - E\alpha\Delta T$$

$$E(y) = \begin{cases} 30e6\ psi, & y < -2in \\ 10e6\ psi, & y \geq -2in \end{cases}, \quad \alpha(y) = \begin{cases} 6.5e{-}6, & y < -2in \\ 13e{-}6, & y \geq -2in \end{cases}$$

$$P^T = \int_A E\alpha\Delta T\, dA, \quad M_y^T = \int_A E\alpha\Delta T\, z\, dA, \quad M_z^T = \int_A E\alpha\Delta T\, y\, dA$$

$$P^T = E_1\alpha_1 \int_{A_1} \Delta T\, dA_1 + E_2\alpha_2 \int_{A_2} \Delta T\, dA_2, \qquad \int_{A_1} \Delta T\, dA_1 = 1.5\int_{-1}^{1.9} \Delta T\, dy$$

$$M_y^T = E_1\alpha_1 \int_{A_1} \Delta T z\, dA_1 + E_2\alpha_2 \int_{A_2} \Delta T z\, dA_2, \qquad \int_{A_2} \Delta T\, dA_1 = 1.5\int_{-1.1}^{-1} \Delta T\, dy$$

$$M_z^T = E_1\alpha_1 \int_{A_1} \Delta T y\, dA_1 + E_2\alpha_2 \int_{A_2} \Delta T y\, dA_2,$$

if $\Delta T = 0$ $\quad P^T = M_y^T = M_z^T = 0$

if $\Delta T = 10y^2 + y^3\ °F$

$P^T$, $M_y^T$, $M_z^T$ come out as constants, which I have calculated in Matlab

$$u_0' = \frac{P + P^T}{E_1 A^*}, \qquad v_0'' = \frac{(M_z - M_z^T) I_{yy}^* + (M_y + M_y^T) I_{yz}^*}{E_1 (I_{yy}^* I_{zz}^* - I_{yz}^{*2})}$$

$$w_0'' = \frac{(M_y + M_y^T) I_{zz}^* + (M_z - M_z^T) I_{yz}^*}{E_1 (I_{yy}^* I_{zz}^* - I_{yz}^{*2})}$$

# Table of Contents

Joshua Oates - HW5 - A331 ..... 1
problem1 ..... 1
for functions, compute constants ..... 1
part e/i ..... 2
output ..... 3
displacements case1 ..... 5
plots of displacements ..... 7
displacement functions ..... 9
functions ..... 9

# Joshua Oates - HW5 - A331

```
clear all
close all
clc
addpath('C:\joshFunctionsMatlab\')

global E1 Ei alpha thing
```

# problem1

```
Ei = [30e6 10e6];
alpha = [6.e-6 13e-6];
E1 = 10e6;
Ei_E1 = Ei./E1;

Ai = [1*1.5 2*1.5];
zip = [0 0];
yip = [-2.5 -1];

b = 1.5;
a1 = 1;
a2 = 2;

Izoi = [a1^3*b/12 a2^3*b/12];
Iyoi = [a1*b^3/12 a2*b^3/12];

Iyzoi = [a1*b*(a1^2+b^2)/12 a2*b*(a2^2+b^2)/12];
thing = joshAdvBeam(Ai,yip,zip,Iyoi,Izoi,Iyzoi,Ei_E1,alpha,Ei);

disp("The location of the modulous weighted centroid in inches is: ( x ,
 "+string(thing.yps)+" , "+string(thing.zps)+" )")
```

```
The location of the modulous weighted centroid in inches is: ( x , -1.9 , 0 )
```

# for functions, compute constants

```
global PT MyT MzT
PT = 1.5679e+04;
```

```
MyT = 0;
MzT = 2.1631e+04;


% PT = 6.2840e+05;
% MyT = 0;
% MzT = 6.3424e+05;
```

# part e/i

```
xdim = 100;
ydim = 3;
zdim = 1.5;

xn = 100;
yn = 50;
zn = 50;

dx = xdim/xn;
dy = ydim/yn;
dz = zdim/zn;

sigMat1 = zeros(xn+1,yn+1,zn+1);
sigMat2 = zeros(xn+1,yn+1,zn+1);
unwrappedi  =1;


for i = 1:xn+1
    for j = 1:yn+1
        for k = 1:zn+1

            x = (i-1)*dx;
            y = (j-1)*dy;
            z = (k-1)*dz;

            y = y-1.1;
            z = z-zdim/2;
            sig = sigxx1(x,y,z);
            sigMat1(i,j,k) = sig;
            SIG1(unwrappedi) = sig;

            sig = sigxx2(x,y,z);
            sigMat2(i,j,k) = sig;
            SIG2(unwrappedi) = sig;

            X(unwrappedi) = x;
            Y(unwrappedi) = y;
            Z(unwrappedi) = z;

            unwrappedi = unwrappedi+1;
        end
    end
end
```

```
% without temp
sigMag1 = abs(sigMat1);
[sigMax1,I1] = max(sigMag1,[],'all');
[i1,j1,k1] = ind2sub(size(sigMag1),I1);


x1 = (i1-1)*dx;
y1 = (j1-1)*dy;
z1 = (k1-1)*dz;

% y1 = y1+thing.yps;
y1 = y1 - 1.1;
z1 = z1-zdim/2;

% with temp
sigMag2 = abs(sigMat2);
[sigMax2,I2] = max(sigMag2,[],'all');
[i2,j2,k2] = ind2sub(size(sigMag2),I2);


x2 = (i2-1)*dx;
y2 = (j2-1)*dy;
z2 = (k2-1)*dz;

y2 = y2 - 1.1;
z2 = z2-zdim/2;
```

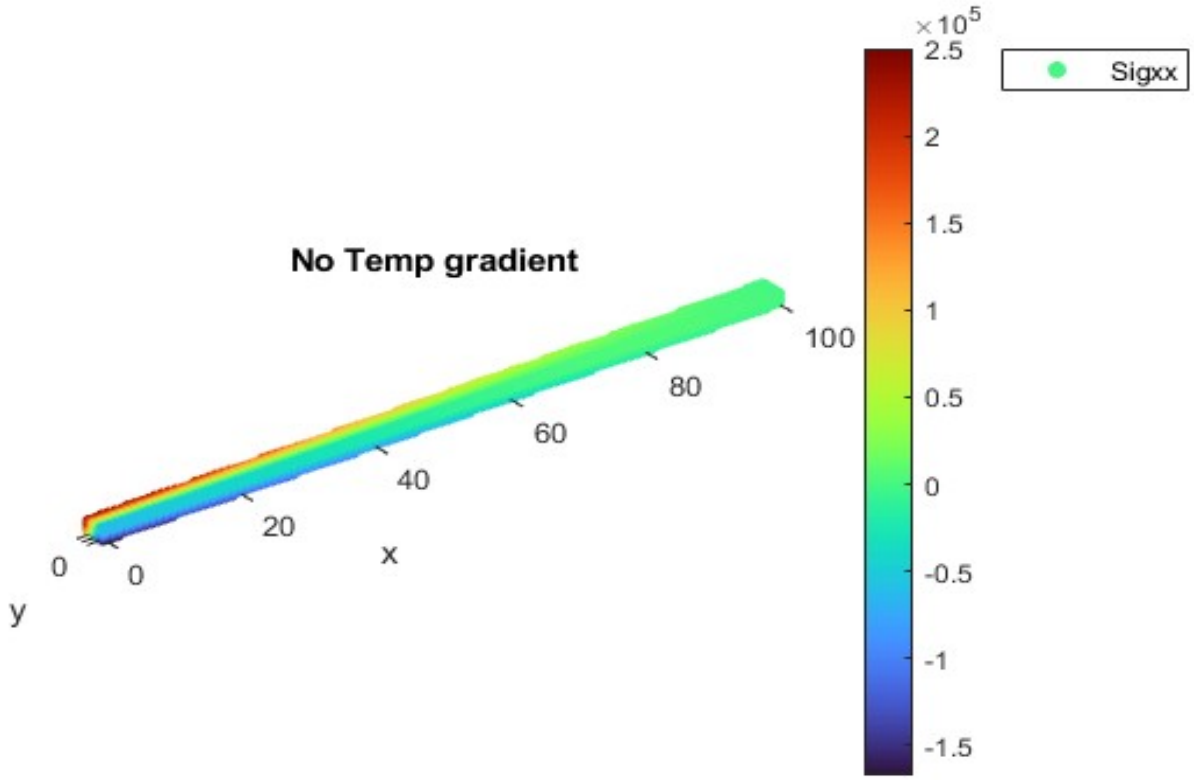
# output

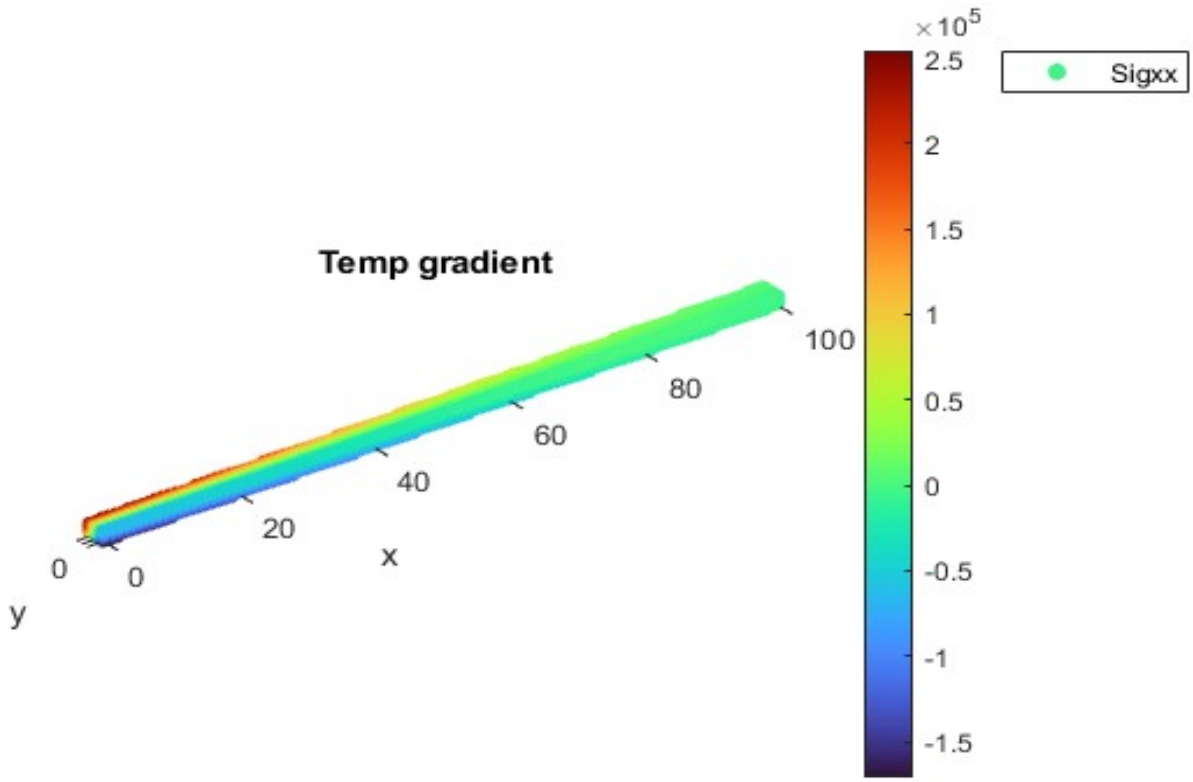
```
disp("I calculated the sigxx at many discrete locations in the beam and found
 the max to be the following:")
% without temp %%%%%%%
disp("If the temperature gradient is zero:")
close all
disp("Sig Max in psi is: "+string(sigMax1))
disp("Sig Max in inches occurs at: ( "+string(x1)+" , "+string(y1)+" ,
 "+string(z1)+" )")
% disp("Index: ( "+string(i1)+" , "+string(j1)+" , "+string(k1)+" )")

figure
scatter3(X,Y,Z,20,SIG1,'filled')
colormap('turbo')
colorbar
title("No Temp gradient")
legend('Sigxx')
xlabel('x')
ylabel('y')
zlabel('z')
axis('equal')


% with temp %%%%%%%%
```

```
disp("If the temperature gradient is applied: ")
disp("Sig Max in psi is: "+string(sigMax2))
disp("Sig Max in inches occurs at: ( "+string(x2)+" , "+string(y2)+" ,
 "+string(z2)+" )")
% disp("Index: ( "+string(i2)+" , "+string(j2)+" , "+string(k2)+" )")

figure
scatter3(X,Y,Z,20,SIG2,'filled')
colormap('turbo')
colorbar
title("Temp gradient")
legend('Sigxx')
xlabel('x')
ylabel('y')
zlabel('z')
axis('equal')
```

*I calculated the sigxx at many discrete locations in the beam and found the max to be the following:*

*If the temperature gradient is zero:*
*Sig Max in psi is: 250337.9416*
*Sig Max in inches occurs at: ( 0 , 1.9 , 0.75 )*
*If the temperature gradient is applied:*
*Sig Max in psi is: 254419.6391*
*Sig Max in inches occurs at: ( 0 , 1.9 , 0.75 )*






# displacements case1

```
syms x u(x) v(x) w(x)
L = 100;

% symbolic expressions for forces
MySym = (L-x)*1000;
Mz1 = 75*L*x-(225/4)*L^2;
Mz2 = 150*L*x-75*x^2-75*L^2;

% boundry conditions
eqn4 = u(0) == 0;
eqn5 = v(0) == 0;
eqn6 = w(0) == 0;

dv = diff(v,x);
dw = diff(w,x);
eqn7 = dv(0) == 0;
eqn8 = dw(0) == 0;

% no heat, base of beam
eqn1 = diff(u,x) == 0;
eqn2 = diff(w,x,2) == ( (MySym)*thing.Izzs+(Mz1)*thing.Iyzs ) /
 E1*(thing.Iyys*thing.Izzs-thing.Iyzs^2);
```

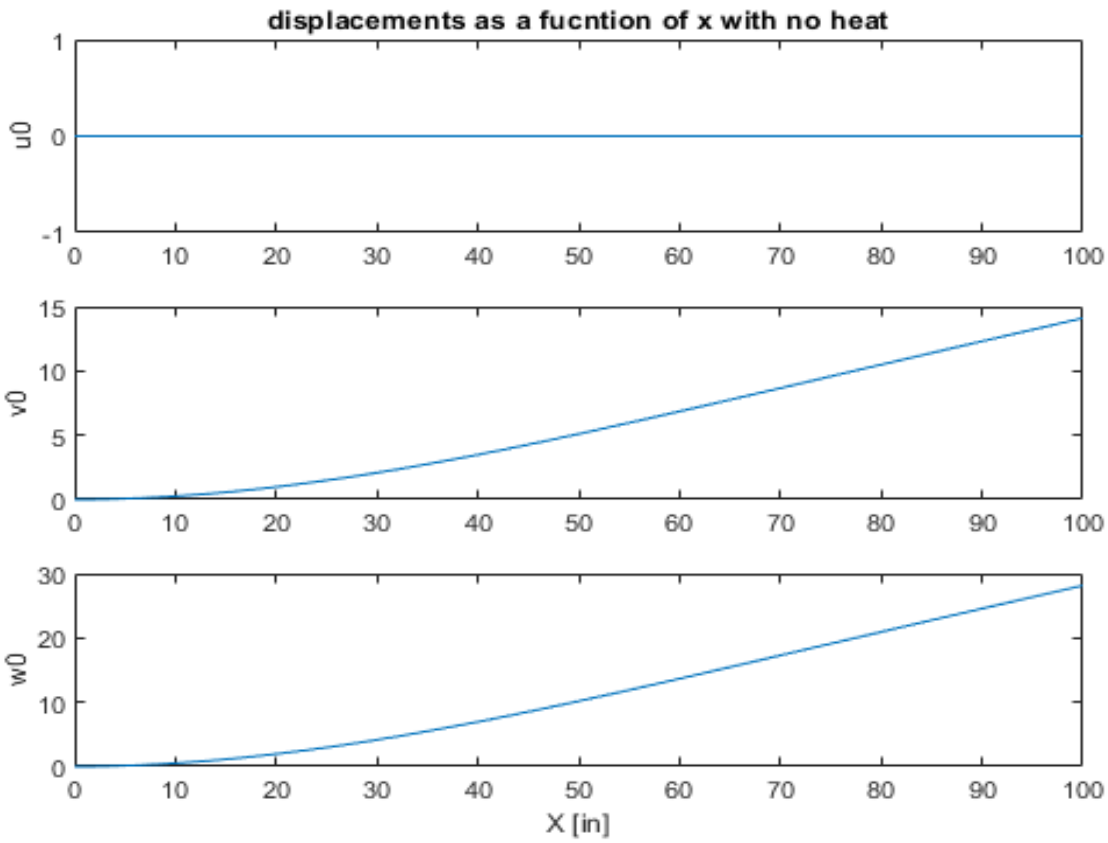
```
eqn3 = diff(v,x,2) == ( (MySym)*thing.Iyzs+(Mz1)*thing.Iyys ) /
 E1*(thing.Iyys*thing.Izzs-thing.Iyzs^2);

sol1Base = dsolve([eqn1,eqn2,eqn3,eqn4,eqn5,eqn6,eqn7,eqn8]);

% heat, base of beam
eqn1 = diff(u,x) == (PT)/(E1*thing.As);
eqn2 = diff(w,x,2) == ( (MySym+MyT)*thing.Izzs+(Mz1-MzT)*thing.Iyzs ) /
 E1*(thing.Iyys*thing.Izzs-thing.Iyzs^2);
eqn3 = diff(v,x,2) == ( (MySym+MyT)*thing.Iyzs+(Mz1-MzT)*thing.Iyys ) /
 E1*(thing.Iyys*thing.Izzs-thing.Iyzs^2);

sol2Base = dsolve([eqn1,eqn2,eqn3,eqn4,eqn5,eqn6,eqn7,eqn8]);


% no heat, tip of beam
eqn1 = diff(u,x) == 0;
eqn2 = diff(w,x,2) == ( (MySym)*thing.Izzs+(Mz2)*thing.Iyzs ) /
 E1*(thing.Iyys*thing.Izzs-thing.Iyzs^2);
eqn3 = diff(v,x,2) == ( (MySym)*thing.Iyzs+(Mz2)*thing.Iyys ) /
 E1*(thing.Iyys*thing.Izzs-thing.Iyzs^2);

eqn4 = u(L/2) == subs(sol1Base.u,x,L/2);
eqn5 = v(L/2) == subs(sol1Base.v,x,L/2);
eqn6 = w(L/2) == subs(sol1Base.w,x,L/2);

dv1 = diff(v,x);
dw1 = diff(w,x);
dv2 = diff(sol1Base.v);
dw2 = diff(sol1Base.w);

eqn7 = dv1(L/2) == subs(dv2,x,L/2);
eqn8 = dw1(L/2) == subs(dw2,x,L/2);


sol1Tip = dsolve([eqn1,eqn2,eqn3,eqn4,eqn5,eqn6,eqn7,eqn8]);

% heat tip of beam
eqn1 = diff(u,x) == (PT)/(E1*thing.As);
eqn2 = diff(w,x,2) == ( (MySym+MyT)*thing.Izzs+(Mz2-MzT)*thing.Iyzs ) /
 E1*(thing.Iyys*thing.Izzs-thing.Iyzs^2);
eqn3 = diff(v,x,2) == ( (MySym+MyT)*thing.Iyzs+(Mz2-MzT)*thing.Iyys ) /
 E1*(thing.Iyys*thing.Izzs-thing.Iyzs^2);

eqn4 = u(L/2) == subs(sol2Base.u,x,L/2);
eqn5 = v(L/2) == subs(sol2Base.v,x,L/2);
eqn6 = w(L/2) == subs(sol2Base.w,x,L/2);

dv1 = diff(v,x);
dw1 = diff(w,x);
dv2 = diff(sol2Base.v);
dw2 = diff(sol2Base.w);

eqn7 = dv1(L/2) == subs(dv2,x,L/2);
```

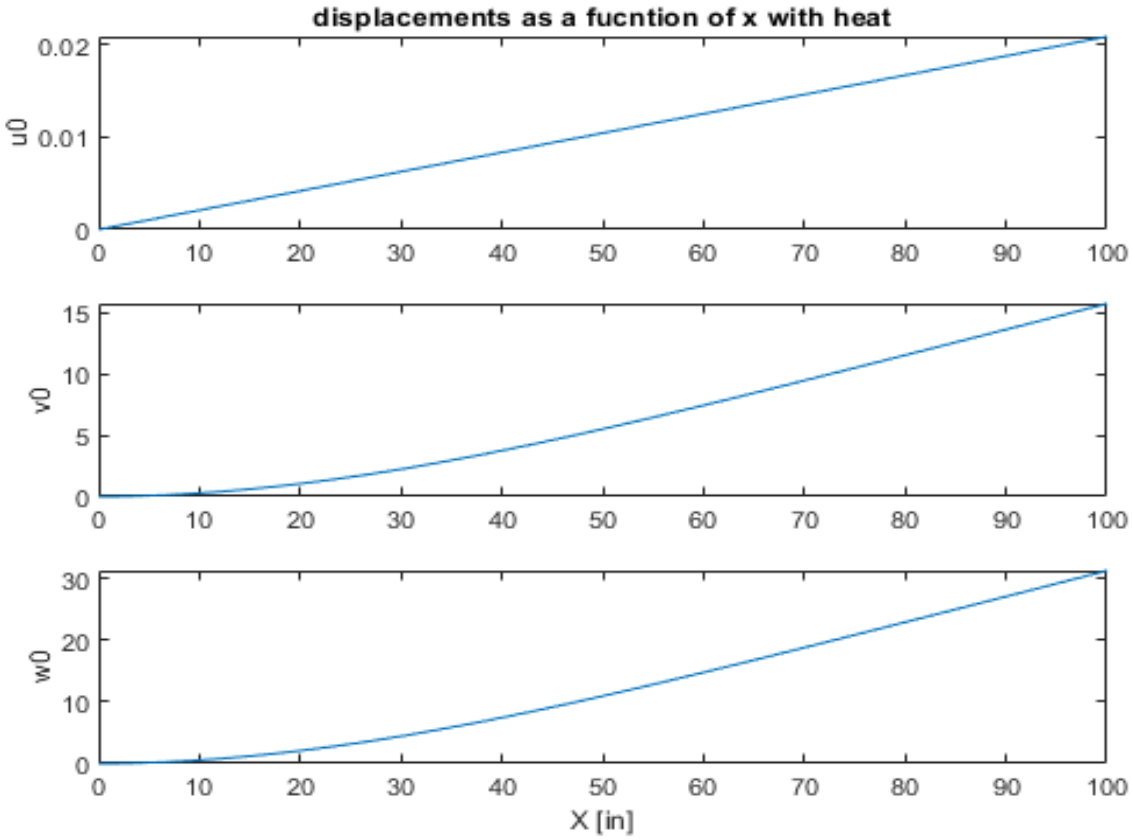
```
eqn8 = dw1(L/2) == subs(dw2,x,L/2);

sol2Tip = dsolve([eqn1,eqn2,eqn3,eqn4,eqn5,eqn6,eqn7,eqn8]);

% to matlab function handles
u1base = matlabFunction(sol1Base.u);
w1base = matlabFunction(sol1Base.w);
v1base = matlabFunction(sol1Base.v);

u2base = matlabFunction(sol2Base.u);
w2base = matlabFunction(sol2Base.w);
v2base = matlabFunction(sol2Base.v);

u1tip = matlabFunction(sol1Tip.u);
w1tip = matlabFunction(sol1Tip.w);
v1tip = matlabFunction(sol1Tip.v);

u2tip = matlabFunction(sol2Tip.u);
w2tip = matlabFunction(sol2Tip.w);
v2tip = matlabFunction(sol2Tip.v);
```

# plots of displacements

```
Xbase = linspace(0,50);
Xtip = linspace(50,100);

figure
subplot(3,1,1)
plot([Xbase,Xtip],zeros(1,length([Xbase,Xtip])))
title("displacements as a fucntion of x with no heat")
ylabel("u0")
subplot(3,1,2)
plot([Xbase,Xtip],[v1base(Xbase),v1tip(Xtip)])
ylabel("v0")
subplot(3,1,3)
plot([Xbase,Xtip],[w1base(Xbase),w1tip(Xtip)])
ylabel("w0")
xlabel("X [in]")



figure
subplot(3,1,1)
plot([Xbase,Xtip],[u2base(Xbase),u2tip(Xtip)])
title("displacements as a fucntion of x with heat")
ylabel("u0")
subplot(3,1,2)
plot([Xbase,Xtip],[v2base(Xbase),v2tip(Xtip)])
ylabel("v0")
subplot(3,1,3)
plot([Xbase,Xtip],[w2base(Xbase),w2tip(Xtip)])
ylabel("w0")
xlabel("X [in]")
```

```
% plot([Xbase,Xtip],[u1base(Xbase),u1tip(Xtip)])
```






# displacement functions

function out = u1vec(x) if x>=50 out = [0,v1tip(x),w1tip(x)]; else out = [0,v1base(x),w1base(x)]; end end

function out = u2vec(x) if x>=50 out = [u2tip(x),v2tip(x),w2tip(x)]; else out = [u2base(x),v2base(x),w2base(x)]; end end

# functions

```
% temp gradients
% function out = DT1(y)
% out = 0;
% end

function out = DT2(y)
out = 10*y^2+y^3;
end

% z Moments
function out = Mz(x)
L = 100;
if x>=50
    out = 150*L*x-75*x^2-75*L^2;
else
    out = 75*L*x-(225/4)*L^2;
```

```
%     out = 7500*x-562500;
end
end


% y moments
function out = My(x)
L = 100;
out = (L-x)*1000;
end


% P force
function out = P(x)
out = 0;
end


% E and alpha as a function of position
function out = E(y)
if y< -2
    out = 30e6;
else
    out = 10e6;
end
end

function out = alphaFun(y)
if y< -2
    out = 6.5e-6;
else
    out = 13e-6;
end
end

% epsilon functions
function out = epsxx1(x,y,z)
global thing E1
out = (P(x))/(E1*thing.As) - (Mz(x))/(E1*thing.Izzs)*y + My(x)/
(E1*thing.Iyys)*z;
end

function out = epsxx2(x,y,z)
global thing PT MyT MzT E1
out = (P(x)+PT) / (E1*thing.As) - (Mz(x)-MzT) / (E1*thing.Izzs)*y +
 (My(x)+MyT) / (E1*thing.Iyys)*z;
end

% sigma functions
function out =  sigxx1(x,y,z)
out = E(y)*(epsxx1(x,y,z));
end

function out =  sigxx2(x,y,z)
```

```
out = E(y)*(epsxx2(x,y,z)-alphaFun(y)*DT2(y));
end
```

*Published with MATLAB® R2022a*

```
function [out] =
 joshAdvBeam(Ai,yi_prime,zi_prime,Iyoiyoi,Izoizoi,Iyoizoi,Ei_E1,alphai,Ei)
% this function takes several



arguments
    Ai {mustBeReal}
    yi_prime {mustBeReal}
    zi_prime {mustBeReal}
    Iyoiyoi {mustBeReal}
    Izoizoi {mustBeReal}
    Iyoizoi {mustBeReal}
    Ei_E1 {mustBeReal}
    alphai {mustBeReal} = nan
    Ei {mustBeReal} = nan
end

A = Ai;
yp = yi_prime;
zp = zi_prime;
Iz0 = Izoizoi;
Iy0 = Iyoiyoi;
Iyz0 = Iyoizoi;

n = length(A);
if length(yp) ~= n  | length(zp) ~= n  | length(Iz0) ~= n  | length(Iy0) ~= n
 | length(Iyz0) ~= n | length(Ei_E1) ~= n
    throw(MException('joshAdvBeam:invalidInput','At least one of the input
 vectors is not the correct length'))
end


% Ai*(Ei/E1)
AE_E1 = Ei_E1.*Ai;
% A*
As = sum(AE_E1);

% A*(E/E1)*y'
AE_E1yp = AE_E1.*yp;
% y'*
yps = sum(AE_E1yp)/As;


% Ai*(Ei/E1)*zi'
AE_E1zp = AE_E1.*zp;
% z'*
zps = sum(AE_E1zp)/As;

% yy
% (Ei/E1)*(Iyoiyoi+Ai'*zi'^2)
var1 = (Ei_E1.*(Iy0+A.*zp.^2));
```

```
% I*y'y'
Iyps = sum(var1);
% I*yy = I*y'y' - A*(z'*)^2
Iys = Iyps - As.*zps.^2;

% zz
var2 = (Ei_E1.*(Iz0+A.*yp.^2));
Izps = sum(var2);
Izs = Izps - As.*yps.^2;

% yz
var3 = (Ei_E1.*(Iyz0+A.*zps.*yps));
Iyzps = sum(var3);
Iyzs = Iyzps - As.*zps.*yps;

% y and z
y = yp-yps;
z = zp-zps;

out.y = y;
out.z = z;
out.As = As;
out.yps = yps;
out.zps = zps;

% out.Iyyps = Iyps;
out.Iyys = Iys;

% out.Izzps = Izps;
out.Izzs = Izs;

% out.Iyzps = Iyzps;
out.Iyzs = Iyzs;

if (~isnan(alphai)) & (~isnan(Ei))

    if length(alphai) ~= n | length(Ei) ~= n
        throw(MException('joshAdvBeam:invalidInput','Either alphai or Ei is
 the wrong length'))
    end
    E = Ei;

    EalphaA = E.*alphai.*A;
    EalphaAy = E.*alphai.*A.*y;
    EalphaAz = E.*alphai.*A.*z;

    PT_DT = sum(EalphaA);
    Mz_DT = sum(EalphaAy);
    My_DT = sum(EalphaAz);

    out.PT_DT = PT_DT;
    out.MzT_DT = Mz_DT;
    out.MyT_DT = My_DT;
end
```

```
end
```

```
Error using joshAdvBeam
Invalid argument list. Function requires 7 more input(s).
```

*Published with MATLAB® R2022a*

```
% constants for HW5
clear all
close all
clc

syms y z
E1 = 30e6;
E2 = 10e6;
a1 = 6.5e-6;
a2 = 13e-6;
DT = 10*y^2+y^3;

PT =
 int(int(E1*a1*DT,y,-1.1,-.1),z,-.75,.75)+int(int(E2*a2*DT,y,-.1,1.9),z,-.75,.75);

MyT =
 int(int(E1*a1*DT*z,y,-1.1,-.1),z,-.75,.75)+int(int(E2*a2*DT*z,y,-.1,1.9),z,-.75,.75);

MzT =
 int(int(E1*a1*DT*y,y,-1.1,-.1),z,-.75,.75)+int(int(E2*a2*DT*y,y,-.1,1.9),z,-.75,.75);

PT = double(vpa(PT));
MyT = double(vpa(MyT));
MzT = double(vpa(MzT));

disp("If the temperature is applied then: ")
disp("PT = "+string(PT)+" lbs")
disp("MyT = "+string(MyT)+" lbs-in")
disp("MzT = "+string(MzT)+" lbs-in")
```

```
If the temperature is applied then:
PT = 6284.005 lbs
MyT = 0 lbs-in
MzT = 6342.4433 lbs-in
```

*Published with MATLAB® R2022a*