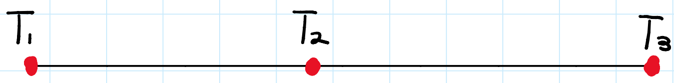
Homework 5

Thursday, February 23, 2023 9:03 PM



$$\int_{x_1}^{x_3} \left(k \frac{\partial^2 T}{\partial x^2} + Q\right) w \, dx = 0$$

using u-v rule:

$$u'w' = (u'w)' - u''w$$

$$\int_{x_1}^{x_3} \left(k\left((T'w)' - T'w'\right) + Qw\right) dx = 0$$

By divergence theorem:

$$\int_{x_1}^{x_2} (T'w)' \, dx = T'w \Big|_{x_1}^{x_2}$$

This then lead us to the reduced weak form:

$$\int_{x_1}^{x_2} \left(k\left(T'w \Big|_{x_1}^{x_2}\right) - kT'w + Qw \right) dx = 0$$

$$kT'w\Big|_{x_1}^{x_2} - k\int_{x_1}^{x_2} (T'w)\,dx + \int_{x_1}^{x_2} (Tw)\,dx = 0$$

We then have:

$$u = N_1T_1 + N_2T_2 + N_3T_3 \qquad w = N_1w_1 + N_2w_2 + N_3w_3$$

$$u' = \frac{\partial u}{\partial x} = N_1'T_1 + N_2'T_2 + N_3'T_3 \qquad w' = N_1'w_1 + N_2'w_2 + N_3'w_3$$

We examine the first term

First Term:

$$ku'w\Big|_{x_1}^{x_3} = T'(x_3)w(x_3) - T'(x_1)w(x_1)$$

$$= k \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} -T_1' \\ 0 \end{bmatrix}$$

$$= K\,[w_1 \;\; w_2 \;\; w_3] \begin{bmatrix} -T_1' \\ 0 \\ T_3' \end{bmatrix}$$

Second Term:

$$K\int_{x_1}^{x_3} (u'w)\,dx = K\int_{x_1}^{x_3} K\,(N_1'T_1 + N_2'T_2 + N_3'T_3)(N_1w_1 + N_2'w_2 + N_3'w_3)\,dx$$

$$= K[w_1 \;\; w_2 \;\; w_3]\left(\int_{x_1}^{x_3} \begin{bmatrix} N_1' \\ N_2' \\ N_3' \end{bmatrix} [N_1' \;\; N_2' \;\; N_3'] \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix}\right)$$

Third Term:

$$\int_{x_1}^{x_2} Qw\,dx = \int_{x_1}^{x_2} Q\,(N_1w_1 + N_2w_2 + N_3w_3)\,dx$$

$$= [w_1 \;\; w_2 \;\; w_3]\left[\int_{x_1}^{x_2} \begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix} Q\,dx\right]$$

re-assembling the terms, we then have:

$$\underbrace{K\int_{x_1}^{x_3} \begin{bmatrix} N_1'^2 & N_1'N_2' & N_1'N_3' \\ N_1'N_2' & N_2'^2 & N_2'N_3' \\ N_1'N_3' & N_2'N_3' & N_3'^2 \end{bmatrix}}_{\text{Local Stiffness}} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \underbrace{\int_{x_1}^{x_2} \begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix} Q\,dx + K \begin{bmatrix} -T_1' \\ 0 \\ T_3' \end{bmatrix}}_{\text{Local Force Vector}}$$

remains zero for interior elements

We have a boundary condition on both sides:

$$T'(x=0) = 0$$

$$T'(x = 12.5\,cm) = -\frac{h}{K}(T-30)$$

$$T'(x = 12.5\,\text{cm}) = -\frac{h}{k}(\underbrace{T}_{T(x=12.5\,\text{cm})} - 30)$$

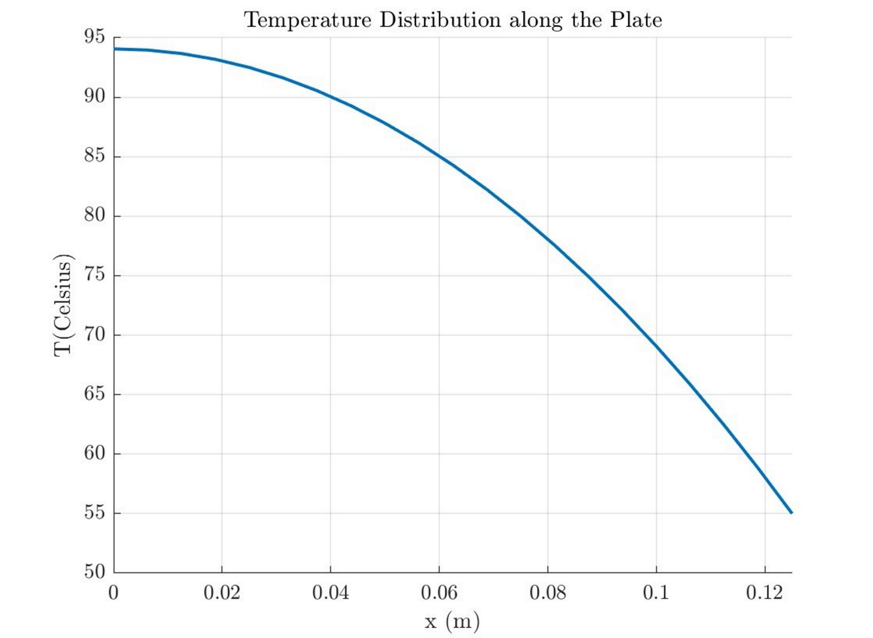
$$N_1' N_3'(T_1) + N_2' N_3'(T_2) + N_3'^2(T_3) = \int_{x_1}^{x_3} N_3 Q\,dx + k T_3'$$

$$N_1' N_3'(T_1) + N_2' N_3'(T_2) + N_3'^2(T_3) = \int_{x_1}^{x_3} N_3 Q\,dx - \cancel{k}\left(\frac{h}{\cancel{k}}(T_3 - 30)\right)$$

$$N_1' N_3'(T_1) + N_2' N_3'(T_2) + N_3'^2(T_3) - hT_3 = \int_{x_1}^{x_3} N_3 Q\,dx + h \cdot 30$$

$$\left[N_1' N_3'(T_1) + N_2' N_3'(T_2) + T_3\underbrace{(N_3'^2 + h)} = \int_{x_1}^{x_3} N_3 Q\,dx + \underbrace{h \cdot 30} \right]$$

We modify the matrix and boundary conditions as follow



```
%AE 510 Class code for live lecture
%Author: Your instructor

set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaultLegendInterpreter','latex')
clc
clear
close all

Len = 12.5/100;%length of the bar
k = 0.8; %W/m - C
Q = 4000; %W/m^3

nelem =10;%number of elements (keep it an even number for this problem)
h = 20; %W/m^3*C
T_amb = 30; %Celsius

%%%%%%%%%%%PREPROCESSING%%%%%%%%%%
%coordinate matrix [x,y] for each node
co = [0 : Len/(2*nelem): Len]';

%element-node connectivity matrix, length, area, modulus
e = [];
for i = 1:2:2*nelem
    temp = [i,i+1,i+2];
    e = [e;temp];
end

Nel = size(e,1);%number of elements
Nnodes = size(co,1); %number of nodes
nne = 3; %number of nodes per element
dof = 1; %degree of freedom per node

%%%%%%%%%%%PREPROCESSING END%%%%%%%%%%


%%%Generic block: Initializes global stiffness matrix 'K' and force vector 'F'
K = zeros(Nnodes*dof,Nnodes*dof);
F = zeros(Nnodes*dof,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%Assemble Global system - generic FE code
for A = 1:Nel

    syms x
    x_co = co(e(A,:),:);

    N_1 = ((x - x_co(2))*(x - x_co(3)))./((x_co(1) - x_co(2))*(x_co(1) -
 x_co(3)));
    N_2 = ((x - x_co(1))*(x - x_co(3)))./((x_co(2) - x_co(1))*(x_co(2) -
 x_co(3)));
```

```
    N_3 = ((x - x_co(1))*(x - x_co(2)))./((x_co(3) - x_co(1))*(x_co(3) -
 x_co(2)));

    dN_1 = diff(N_1,x);
    dN_2 = diff(N_2,x);
    dN_3 = diff(N_3,x);
    localstiffness = [dN_1.^2, dN_1*dN_2, dN_1*dN_3;
                      dN_1*dN_2, dN_2.^2, dN_2*dN_3;
                      dN_1*dN_3, dN_2*dN_3,dN_3.^2];

    localstiffness = double(int(k.*localstiffness,x,x_co(1),x_co(3)));

    localforce = [N_1;N_2;N_3];
    localforce = double(int(localforce.*Q,x,x_co(1),x_co(3)));

    %DONT TOUCH BELOW BLOCK!! Assembles the global stiffness matrix, Generic
 block which works for any element

    for B = 1: nne
        for i = 1: dof
            nK1 = (e(A, B)-1)*dof+i;
            nKe1 = (B-1)*dof+i;
            F(nK1) = F(nK1) + localforce(nKe1);
            for C = 1: nne
                for j = 1: dof
                    nK2 = (e(A, C)-1)*dof+j;
                    nKe2 = (C-1)*dof+j;
                    K(nK1, nK2) = K(nK1, nK2) + localstiffness(nKe1, nKe2);
                end
            end
        end
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

end


%%%%%%%%%%%%%%%%%%%BOUNDARY CONDITIONS%%%%%%%%%%%%%%%%%%%%%%%%

%external forces

F(end) = F(end) + h*T_amb; %given x- component of force in node 2
K(end,end) = K(end,end) + h;

%Apply displacement BC by eliminating rows and columns of nodes 3-4
 (corresponding to
%degrees of freedom 5 to 8) - alternative (and more generic method) is the
 penalty approach, or
%static condensation approach - see later class notes

% deletedofs = [1];%first nodes
```

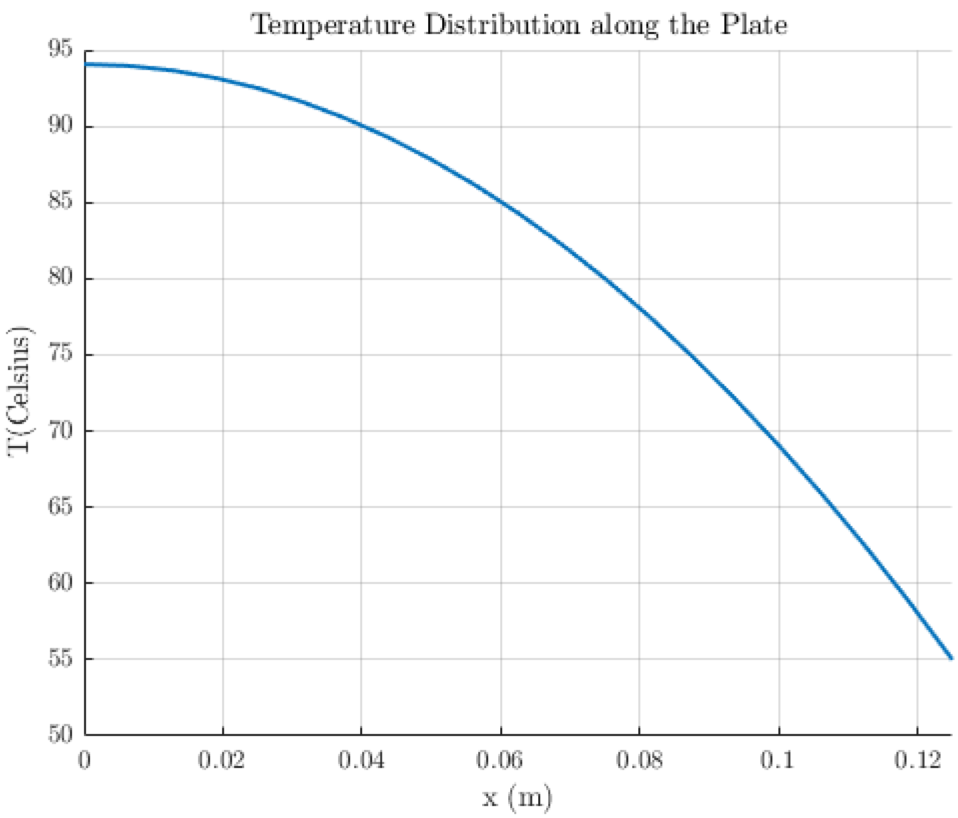
```
% K(deletedofs,:) = [];
% K(:,deletedofs) = [];
% F(deletedofs,:) = [];

%%%%%%%%%%%%%%%%%%%BOUNDARY CONDITIONS END%%%%%%%%%%%%%%%%%%%%%%%


%solve for displacement unknowns (uk)
uk = K\F;

%expand u to include deleted displacement bcs
% u = ones(Nnodes*dof,1);
% u(deletedofs) = 0;
% I = find(u == 1);
% u(I) = uk;


figure()
hold on
plot(co,uk,'LineWidth',1.5)
title('Temperature Distribution along the Plate')
xlabel('x (m)')
ylabel('T(Celsius)')
xlim([0,Len])
grid on
```



Published with MATLAB® R2022b