Assembling the global stiffness matrix

1 solution submitted (max: 2) | View my solutions

Open Item in MATLAB Online

In this problem, a plane truss has been defined. The definition is effected through the nodal x coordinates contained in the vector x and y coordinates in y. Additionally, we have also defined a element connectivity matrix $\mathrm{conn}(1:\mathrm{nelem}, 1:\mathrm{ndof}*\mathrm{nnod})$, where nelem is the number of elements defined (in our case 9), while nnod and ndof are the number of nodes per element (in case of truss elements 2) and the number of degrees of freedom per node (also 2 for a space truss).

The external loads are applied at nodes 2, 3 and 5 in the negative y-direction. The magnitude of each load is $20\ \mathrm{kN}$.

Every truss element is made of steel with Young's modulus $E = 201\ \mathrm{GPa}$ and cross sectional area of $10 \times 10^{-4}\ \mathrm{m}^2$.

First, on a rough sheet of paper, sketch the plane truss using the information given. Make sure you have the global node and element numberings exactly according to what is specified. Also, apply the external forces at the nodes and directions in which they are specified.

Next, assume that the truss rests on roller supports at $(-8, 0)$ and $(8, 0)$.

Form the destination array in the matrix dest and assemble the global stiffness in Kg and external force vector in F. Do not forget to first initialise Kg to a zero matrix of the appropriate size.

Use force units of N and length units of m.

```matlab
% problem 1 for quiz 2, plane truss
EA=210e09*10e-04;
% nodal coordinates
x=[-8 -4 0 0 4 8];
y=[0 2 4 0 2 0];
tot_nodes=length(x);
% element connectivity
conn = [1 2;1 4;4 2;2 3;4 3;3 5;4 5;4 6;5 6];
nelem= size(conn, 1);
% form destination array
dest = [1 2 3 4;1 2 7 8;7 8 3 4;3 4 5 6;7 8 5 6;5 6 9 10;7 8 9 10;7 8 11
12;9 10 11 12;];
hi=size(dest);
% form element stiffness and assemble
stiff_size =tot_nodes ; % size of the global stiffness
Kg = zeros(2*tot_nodes,2*tot_nodes ); % initialise global stiffness to
all zeros
for ielem=1:nelem
      n1 = conn(ielem, 1);
   n2 = conn(ielem, 2);

   % Node coordinates
   x1 = x(n1); y1 = y(n1);
   x2 = x(n2); y2 = y(n2);

   % Element length
   Lelem = sqrt((x2 - x1)^2 + (y2 - y1)^2);

   % Direction cosines
   cos_theta = (x2 - x1) / Lelem;
```

```
    sin_theta = (y2 - y1) / Lelem;

    % Local stiffness matrix in local coordinates
    Ke_local = (EA / Lelem) * [1 0 -1 0;
                                0 0  0 0;
                               -1 0  1 0;
                                0 0  0 0];

     % Transformation matrix
    T = [ cos_theta, sin_theta, 0,         0;
         -sin_theta, cos_theta, 0,         0;
          0,         0,         cos_theta, sin_theta;
          0,         0,        -sin_theta, cos_theta];

    % Element stiffness in global coordinates
    Ke_global = T' * Ke_local * T;

    % Destination DOFs
    dest = [2*n1-1, 2*n1, 2*n2-1, 2*n2];

    % Assemble into global stiffness matrix
    Kg(dest, dest) = Kg(dest, dest) + Ke_global;


end
% assemble force
P=-20e03;
F = zeros(2 * tot_nodes, 1);
% Apply loads at nodes 2, 3, and 5 in negative y-direction
F(4) = P;    % Node 2, y-direction
F(6) = P;    % Node 3, y-direction
F(10) = P;   % Node 5, y-direction


Kg
```