Applying boundary conditions

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

Open Item in MATLAB Online

Again, a plane truss is defined through the vectors `x` and `y` as in Prob. 1. Similarly, the connectivity is also defined in `conn`. As before, sketch the geometry on a piece of paper. The ground on which the truss rests is perpendicular to the x axis.

A force P of 1000 kN is applied at node 1 in the positive y direction.

Nodes 2 and 3 are hinged. All truss elements are made of steel with $E = 210$ GPa and area of cross section $A = 6 \times 10^{-4}$ m^2.

In addition to the load P, node 1 has an additional displacement of 50 mm in the negative x direction caused by a gradual sinking of the support into the ground.

Form the destination array `dest` and assemble the global stiffness `Kg`, like in the previous problem. Also form the global force vector `F`.

Finally modify `Kg` and `F` for the given boundary conditions. Note that, after modification, `Kg` must be non-singular and hence, invertible. In other words `eig(Kg)` should yield all non-zero eigenvalues.

Use N for force units and m for length.

```
% problem 2 for quiz 2, plane truss, boundary conditions
EA=210e09*6e-04;
% nodal coordinates
x=[0 3 0];
y=[0 4 4];
nnode=length(x);
% element connectivity
conn = [2 1;1 3];
nelem= size(conn, 1);
% destination array
for ielem=1:nelem
   n1=conn(ielem,1); n2=conn(ielem,2);
   dest(ielem,:) = [2*n1-1;2*n1;2*n2-1;2*n2];
end
% form element stiffness and assemble
tot_ndof = 2*nnode; %size of the global stiffness
Kg = zeros(tot_ndof); % initialise global stiffness to zeros

for ielem=1:nelem
  n1=conn(ielem,1); n2=conn(ielem,2);
   dx=x(n2)-x(n1); dy=y(n2)-y(n1);
   L=sqrt(dx^2+dy^2);
   c=dx/L; s=dy/L;
   Ke = (EA/L)*[ c^2    c*s   -c^2  -c*s;
                 c*s    s^2   -c*s  -s^2;
                -c^2   -c*s    c^2   c*s;
                -c*s   -s^2    c*s   s^2];
   dofs = dest(ielem,:);
   Kg(dofs,dofs) = Kg(dofs,dofs) + Ke;
end
```

```
% % form rhs force vector
P=1000e03;
F = zeros(tot_ndof,1);
F(2)=P;


% % apply boundary conditions, modify force and global stiffness store
in Kgprime and Fprime
Fprime=F;
Kgprime=Kg;
% === impose essential BCs, keep sizes 6x6 and 6x1 ===
% known displacements: u1x = -0.05 m; node 2 and node 3 fully fixed
bc_dofs  = [1 3 4 5 6];
u_presc  = zeros(tot_ndof,1);
u_presc(1) = -0.05;

% move known displacements to RHS using ORIGINAL Kg (not yet modified)
Fprime = F - Kg(:,bc_dofs)*u_presc(bc_dofs);

% modify stiffness and put prescribed values into Fprime at BC dofs
Kgprime = Kg;
for d = bc_dofs
   Kgprime(d,:) = 0;
   Kgprime(:,d) = 0;
   Kgprime(d,d) = 1;
   Fprime(d)    = u_presc(d);
end
```