```matlab
% Author: Nhan Nguyen
% FEM - Direct stiffness method code for structural analysis of bar element
% truss assembly

clear, clc

% Given values for each element
A = [1/sqrt(2); 1; 1];
E = [10^6; 10^6; 10^6];
L = [sqrt(2); 1; sqrt(2)];
theta = [135; 0; -135];
% Define connection matrix (ex: row 1 indicates element 1, connected by
% nodes 1,2)
CM = [1 4; 2 4;3 4];


% Define number of elements and nodes
numElements = 3;
numNodes = 4;
numDOFperNode = 2; % For rotational bar element DOF = 2
numDOF = numNodes * numDOFperNode;

% Initialize global stiffness matrix
K = zeros(numDOF, numDOF);

% Loop through each element to calculate and assemble local stiffness matrices
% Initialize a cell array to store local stiffness matrices
localStiffnessMatrices = cell(numElements, 1);
dof_indices = cell(numElements, 1);


for i = 1:numElements
    % Extract element properties
    E_i = E(i);
    A_i = A(i);
    L_i = L(i);
    theta_i = theta(i);
    c = cosd(theta_i);
    s = sind(theta_i);
    % Calculate local stiffness matrix for the current element
    k_i = E_i * A_i / L_i * [
        c^2 c*s -c^2 -c*s;
        0 s^2 -c*s -s^2;
        0 0 c^2 c*s;
        0 0 0 s^2];

    % Store the local stiffness matrix in the cell array
    localStiffnessMatrices{i} = k_i;

    % Compute the global nodes corresponding to the element
    nodes = CM(i, :);

    % Convert local node numbers to global node indices

    dof_indices{i} = [2*nodes(1)-1, 2*nodes(1), 2*nodes(2)-1, 2*nodes(2)];

    % Add the local stiffness matrix to the global stiffness matrix
    K(dof_indices{i}, dof_indices{i}) = K(dof_indices{i}, dof_indices{i}) + k_i;
end
```

```
% Define external forces vector
F = zeros(numDOF, 1);
F(8) = -10000; % Applied force at node 4

% Define boundary conditions:
D = zeros(numDOF, 1); % displacement vector

bc = [1,2,3,4,5,6];

% Apply boundary conditions
K_reduced = K;
F_reduced = F;
for i = length(bc):-1:1
    K_reduced(bc(i), :) = [];
    K_reduced(:, bc(i)) = [];
    F_reduced(bc(i)) = [];
end

% Solve for displacements symbolically
D_reduced = K_reduced \ F_reduced;

% Substitute d_reduced back into d
D(setdiff(1:numDOF, bc)) = D_reduced;

% Solve for F
F = K * D;

% Part a: Print the Global Stiffness Matrix
disp('Global Stiffness Matrix (K):');
disp(K);

% Part b: Print the Reduced Stiffness Matrix
disp('Reduced Stiffness Matrix');
disp(K_reduced);

% Part c: Solve for unknown displacements
disp('Displacement Matrix (d)');
disp(D);

% Part d: Solve for unknown reactions:
disp('Forces matrix (F)');
disp(F);

%Display internal forces for each element
for i = 1:numElements
    fprintf('Internal forces matrix (F) for Element %d:\n', i);
    disp(localStiffnessMatrices{i} * D(dof_indices{i}));
end
```

```
Global Stiffness Matrix (K):
   1.0e+06 *

  Columns 1 through 7

    0.2500   -0.2500         0         0         0         0   -0.2500
         0    0.2500         0         0         0         0    0.2500
         0         0    1.0000         0         0         0   -1.0000
         0         0         0         0         0         0         0
         0         0         0         0    0.3536    0.3536   -0.3536
         0         0         0         0         0    0.3536   -0.3536
         0         0         0         0         0         0    1.6036
```

```
         0         0         0         0         0         0         0

  Column 8

    0.2500
   -0.2500
         0
         0
   -0.3536
   -0.3536
    0.1036
    0.6036

Reduced Stiffness Matrix
   1.0e+06 *

    1.6036    0.1036
         0    0.6036

Displacement Matrix (d)
         0
         0
         0
         0
         0
         0
    0.0011
   -0.0166

Forces matrix (F)
   1.0e+04 *

   -0.4410
    0.4410
   -0.1070
         0
    0.5480
    0.5480
         0
   -1.0000

Internal forces matrix (F) for Element 1:
   1.0e+03 *

   -4.4096
    4.4096
    4.4096
   -4.1421

Internal forces matrix (F) for Element 2:
   1.0e+03 *

   -1.0700
         0
    1.0700
         0

Internal forces matrix (F) for Element 3:
   1.0e+03 *

    5.4796
    5.4796
```

```
  -5.4796
  -5.8579
```

Published with MATLAB® R2023b