Stiffness Methods for Systematic Analysis of Structures

(Ref: Chapters 14, 15, 16)

The Stiffness method provides a very systematic way of analyzing determinate and indeterminate structures.

Recall

Force (Flexibility) Method

- Convert the indeterminate structure to a determinate one by removing some unknown forces / support reactions and replacing them with (assumed) known / unit forces.
- Using superposition, calculate the force that would be required to achieve compatibility with the original structure.
- Unknowns to be solved for are usually redundant forces
- Coefficients of the unknowns in equations to be solved are "flexibility" coefficients.

$$[A]x = b$$

- Additional steps are necessary to determine displacements and internal forces
- Can be programmed into a computer, but human input is required to select primary structure and redundant forces.

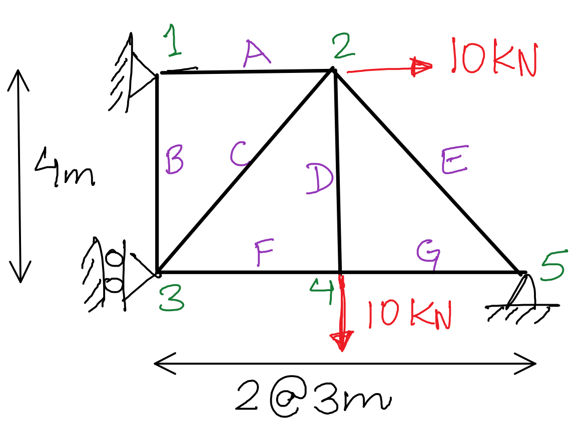
Displacement (Stiffness) Method

- Express local (member) force-displacement relationships in terms of unknown member displacements.
- Using equilibrium of assembled members, find unknown displacements.
- Unknowns are usually displacements
- Coefficients of the unknowns are "Stiffness" coefficients.

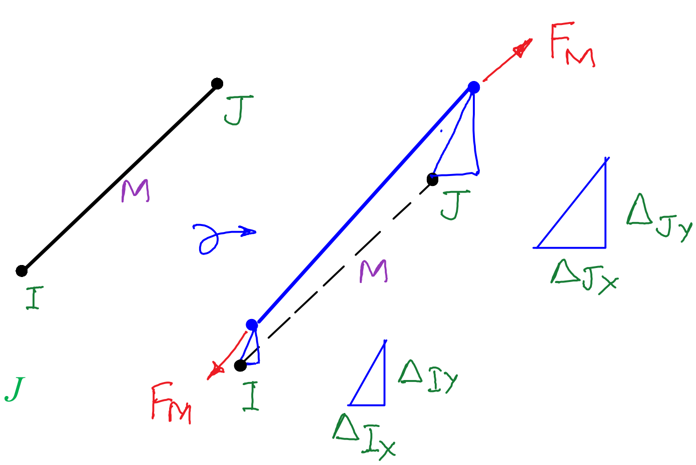
$$[K]d = f$$

- Directly gives desired displacements and internal member forces
- Easy to program in a computer

Example:



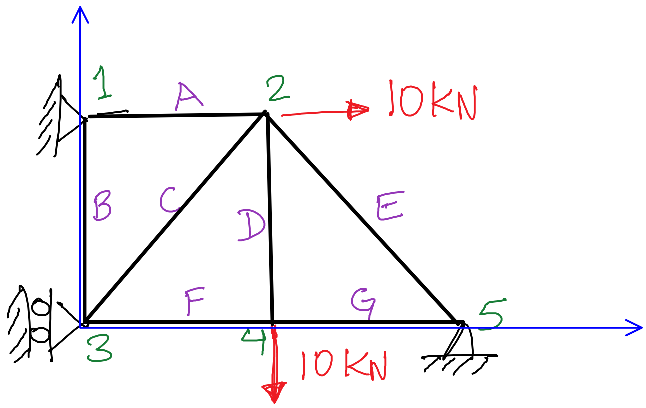
Divide the structure into MEMBERS (A, B, ---) and NODES (1, 2, ---)



Overall idea:

- Express F_M in terms of displacements of I and J
- Assemble ALL members and enforce EQUILIBRIUM to find displacements.

Member and Node Connectivity:

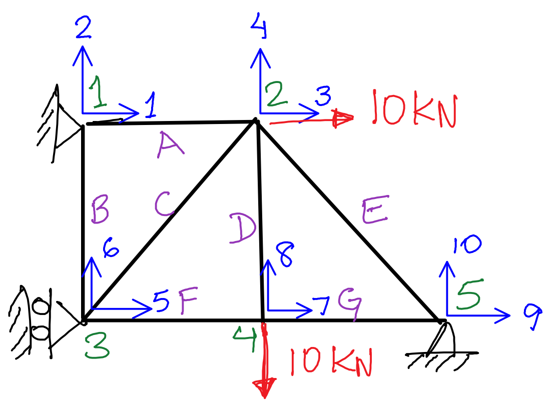


NODE	LOCATIONS
1	(0, 4)
2	(3, 4)
3	(0, 0)
4	(3, 0)
5	(6, 0)

MEMBER CONNECTIVITY

A : 1,2
B : 1,3
C : 3,2
D : 2,4
E : 2,5
F : 3,4
G : 4,5

Degrees of Freedom (Kinematic Indeterminacy)



Associate member displacements with DEGREES OF FREEDOM (DOF) (KINEMATIC INDETERMINACY)

Note :

$$\text{NODE} : \overbrace{[2*\text{NODE}-1 \; ; \; 2*\text{NODE}]}^{\text{DOFS}}$$

1 : [1 ; 2]
2 : [3 ; 4]
3 : [5 ; 6]
4 : [7 ; 8]
5 : [9 ; 10]

Also Note :

FREE DOFS : [3, 4, 6, 7, 8]

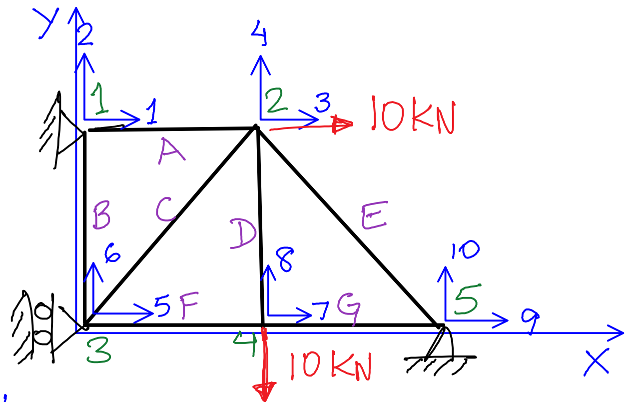
SPECIFIED DOFS : [1, 2, 5, 9, 10]

Global and Local (member) co-ordinate axes

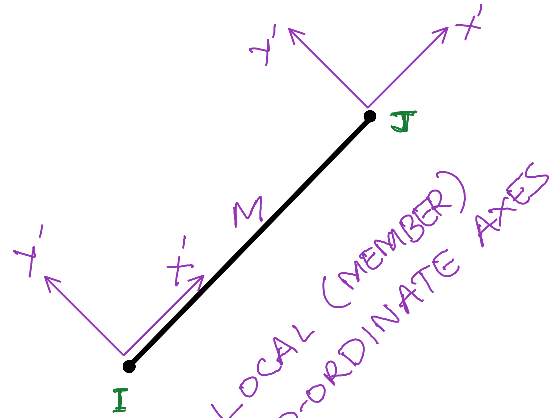
In order to relate:

- Global displacements with Local (member) deformations, and
- Local member forces back to Global force equilibrium,

we need to be able to transform between these 2 co-ordinate axes freely:

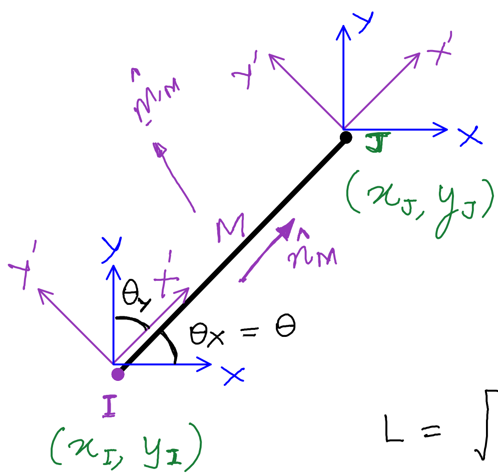


GLOBAL COORDINATE AXES



LOCAL (MEMBER) CO-ORDINATE AXES

Transformation of Vectors (Displacements or Forces) between Global and Local coordinates

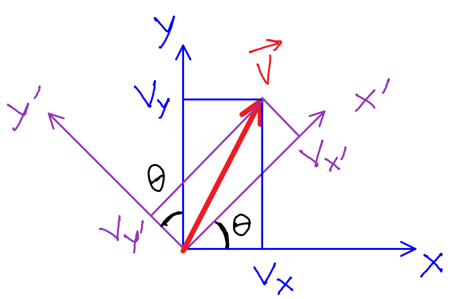


Note: $\cos(\theta_x)$ & $\cos(\theta_y)$ are DIRECTION COSINES for member M.

$$\cos(\theta_x) = \frac{(x_J - x_I)}{L}, \qquad \cos(\theta_y) = \frac{(y_J - y_I)}{L}, \qquad \begin{bmatrix} \cos(\theta_x) \\ \cos(\theta_y) \end{bmatrix} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}$$

$$L = \sqrt{(x_J - x_I)^2 + (y_J - y_I)^2}$$

$$\hat{n}_M = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}$$



$$\vec{V} = \{V_x, V_y\}_{x,y} = \{V_{x'}, V_{y'}\}_{x',y'}$$

Note:

$$V_x = V_{x'}\cos(\theta) - V_{y'}\sin(\theta)$$
$$V_y = V_{x'}\sin(\theta) + V_{y'}\cos(\theta)$$

In matrix form:

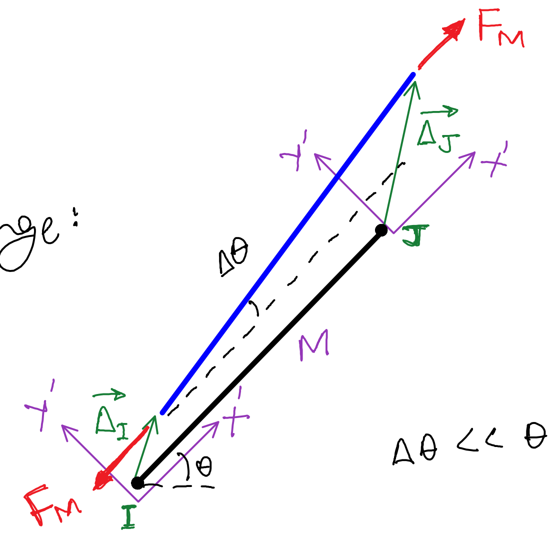
$$\begin{Bmatrix} V_x \\ V_y \end{Bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{\underset{\sim}{T}} \begin{Bmatrix} V_{x'} \\ V_{y'} \end{Bmatrix}$$

Reverse:

$$\begin{Bmatrix} V_{x'} \\ V_{y'} \end{Bmatrix} = \left[\underset{\sim}{T}^{-1} \right] \begin{Bmatrix} V_x \\ V_y \end{Bmatrix} = \left[\underset{\sim}{T}^{T} \right] \begin{Bmatrix} V_x \\ V_y \end{Bmatrix}$$

Local (Member) Force-Displacement Relationships

Assuming $\vec{\Delta}_I$ and $\vec{\Delta}_J$ are small so that the orientation "θ" of the member does NOT change:



Change in length

$$\Delta L = \Delta'_{Jx} - \Delta'_{Ix}$$

$$F_M = \frac{AE}{L} \cdot \Delta L$$

$$= \frac{AE}{L}\left(-\Delta'_{Ix} + \Delta'_{Jx}\right)$$

Thus $\boxed{\text{local}}$ (member) force-displacement relationship:

$$\boxed{\begin{array}{c} I \\ \\ J \\ \\ \end{array}\begin{Bmatrix} -F_M \\ 0 \\ F_M \\ 0 \end{Bmatrix} = \frac{AE}{L}\underbrace{\begin{bmatrix} +1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & +1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{} \begin{Bmatrix} \Delta'_{Ix} \\ \Delta'_{Iy} \\ \Delta'_{Jx} \\ \Delta'_{Jy} \end{Bmatrix}\begin{array}{c} I \\ \\ J \\ \\ \end{array}}$$

MEMBER FORCES IN LOCAL CO-ORDINATES

LOCAL MEMBER STIFFNESS MATRIX $[\underset{\sim}{K}'_M]$

NODE DISPLACEMENTS IN LOCAL CO-ORDINATES

Thus:

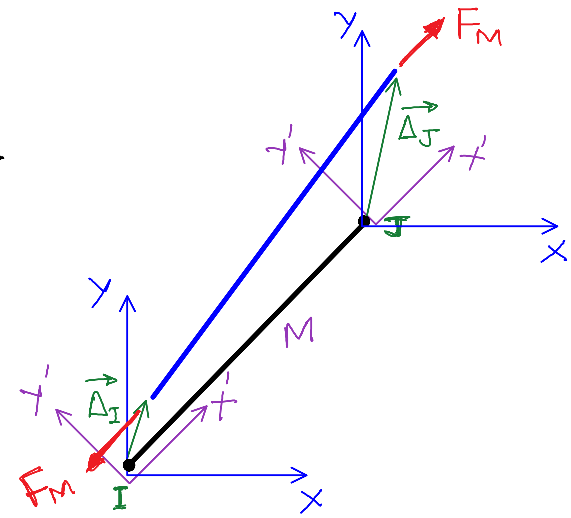
$$\boxed{\underline{q}'_M = \underset{\sim}{K}'_M \; \Delta'_M}$$

These LOCAL (member) force-displacement relationships can be easily established for ALL the members in the truss, simply by using given material and geometric properties of the different members.

ASSEMBLY of LOCAL force-displacement relationships for GLOBAL Equilibrium

The member forces that were expressed in the LOCAL coordinate system, cannot be directly added to one another to obtain GLOBAL equilibrium of the structure.

They must be TRANSFORMED from LOCAL to GLOBAL and then added together to obtain the global equilibrium equations for the structure which will allow us to solve for the unknown displacements.



Note:

$$\overset{\underline{q}_M}{\begin{Bmatrix} F_{Ix} \\ F_{Iy} \\ F_{Jx} \\ F_{Jy} \end{Bmatrix}} = \begin{bmatrix} \underset{\sim}{T}_M & 0 \\ 0 & \underset{\sim}{T}_M \end{bmatrix} \overset{\underline{q}_M'}{\begin{Bmatrix} -F_M \\ 0 \\ F_M \\ 0 \end{Bmatrix}}$$

ie

$$\boxed{\underline{q}_M = \begin{bmatrix} \underset{\sim}{T}_M & 0 \\ 0 & \underset{\sim}{T}_M \end{bmatrix} \underline{q}_M'}$$

$$\begin{Bmatrix} \underline{q}_I \\ \underline{q}_J \end{Bmatrix} = \begin{bmatrix} \underset{\sim}{T}_M & 0 \\ 0 & \underset{\sim}{T}_M \end{bmatrix} \underset{\sim}{K}_M' \begin{Bmatrix} \underline{\Delta}_I' \\ \underline{\Delta}_J' \end{Bmatrix}$$

$$\begin{Bmatrix} \underline{q}_I \\ \underline{q}_J \end{Bmatrix} = \underbrace{\begin{bmatrix} \underset{\sim}{T}_M & 0 \\ 0 & \underset{\sim}{T}_M \end{bmatrix} \underset{\sim}{K}_M' \overbrace{\begin{bmatrix} \underset{\sim}{T}_M^T & 0 \\ 0 & \underset{\sim}{T}_M^T \end{bmatrix}}}_{\underset{\sim}{K}_M} \begin{Bmatrix} \underline{\Delta}_I \\ \underline{\Delta}_J \end{Bmatrix}$$

Thus

$$\boxed{\begin{Bmatrix} \underline{q}_I \\ \underline{q}_J \end{Bmatrix} = \begin{bmatrix} \underset{\sim}{K}_M \end{bmatrix} \begin{Bmatrix} \underline{\Delta}_I \\ \underline{\Delta}_J \end{Bmatrix}}$$

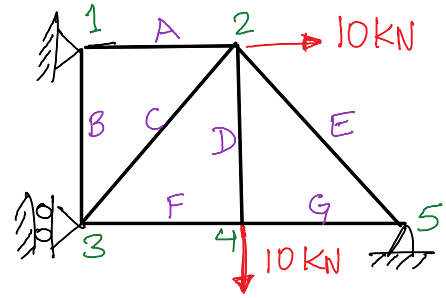
GLOBAL FORCE-DISPLACEMENT RELATIONSHIP FOR MEMBER "M"

where

$$\begin{bmatrix} \underset{\sim}{K}_M \end{bmatrix} = \begin{bmatrix} \underset{\sim}{T}_M & 0 \\ 0 & \underset{\sim}{T}_M \end{bmatrix} \begin{bmatrix} \underset{\sim}{K}_M' \end{bmatrix} \begin{bmatrix} \underset{\sim}{T}_M^T & 0 \\ 0 & \underset{\sim}{T}_M^T \end{bmatrix}$$

ASSEMBLY of LOCAL force-displacement relationships for GLOBAL Equilibrium

Now ALL the member force-displacement relationships can be ASSEMBLED (Added) together to get Global equilibrium:



GLOBAL NODES ↓ MEMBERS →

$$
\begin{array}{c|c}
 & \quad A \qquad\quad B \qquad\quad C \qquad\quad D \qquad\quad E \qquad\quad F \qquad\quad G \qquad\qquad EXT \\
\begin{matrix} 1 \\ \\ 2 \\ \\ 3 \\ \\ 4 \\ \\ 5 \\ \\ \end{matrix} &
-\left[\begin{bmatrix} q_1^A \\ q_2^A \end{bmatrix} + \begin{bmatrix} q_1^B \\ q_3^B \end{bmatrix} + \begin{bmatrix} q_2^C \\ q_3^C \end{bmatrix} + \begin{bmatrix} q_2^D \\ q_4^D \end{bmatrix} + \begin{bmatrix} q_2^E \\ q_5^E \end{bmatrix} + \begin{bmatrix} q_3^F \\ q_4^F \end{bmatrix} + \begin{bmatrix} q_4^G \\ q_5^G \end{bmatrix} \right] + \begin{bmatrix} ? \\ ? \\ \begin{Bmatrix} 10 \\ 0 \end{Bmatrix} \\ ? \\ 0 \\ \begin{Bmatrix} 0 \\ -10 \end{Bmatrix} \\ ? \\ ? \end{bmatrix} = \underline{0}
\end{array}
$$

Note that "q" are forces on members, so to get forces on nodes we must take "-q".
Each one of the 10 equations above must sum to ZERO for global equilibrium.

Further recall that:

$$
\begin{Bmatrix} \underline{q}_I \\ \underline{q}_J \end{Bmatrix} = \underbrace{\begin{bmatrix} \underset{\sim}{K}_{II} & \underset{\sim}{K}_{IJ} \\ \underset{\sim}{K}_{JI} & \underset{\sim}{K}_{JJ} \end{bmatrix}}^{\underset{\sim}{K}_M} \begin{Bmatrix} \underline{\Delta}_I \\ \underline{\Delta}_J \end{Bmatrix}
$$

Thus:

$$
\begin{array}{c|c}
 & \qquad 1 \qquad\qquad\qquad 2 \qquad\qquad\qquad\qquad 3 \qquad\qquad\qquad\qquad 4 \qquad\qquad\qquad 5 \\
\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} &
\begin{bmatrix}
(K_{11}^A + K_{11}^B) & K_{12}^A & K_{13}^B & & \\
K_{21}^A & (K_{22}^A + K_{22}^C + K_{22}^D + K_{22}^E) & K_{23}^C & K_{24}^D & K_{25}^E \\
K_{31}^B & K_{32}^C & (K_{33}^B + K_{33}^C + K_{33}^F) & K_{34}^F & \\
 & K_{42}^D & K_{43}^F & (K_{44}^D + K_{44}^F + K_{44}^G) & K_{45}^G \\
 & K_{52}^E & & K_{54}^G & (K_{55}^E + K_{55}^G)
\end{bmatrix}
\begin{bmatrix} \underline{\Delta}_1 \\ \underline{\Delta}_2 \\ \underline{\Delta}_3 \\ \underline{\Delta}_4 \\ \underline{\Delta}_5 \end{bmatrix}
\end{array}
$$

$$
= \begin{bmatrix} ? \\ ? \\ \begin{Bmatrix} 10 \\ 0 \end{Bmatrix} \\ ? \\ 0 \\ \begin{Bmatrix} 0 \\ -10 \end{Bmatrix} \\ ? \\ ? \end{bmatrix}
$$

These are the $5 \times 2 = 10$ equations representing GLOBAL equilibrium.

Solution of unknown displacements at "free dofs" and reactions at "specified dofs"

$$
1.0e{+}06\ *\
\begin{array}{c|cccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\hline
1 & 0.6667 & 0 & -0.6667 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0.5000 & 0 & 0 & 0 & -0.5000 & 0 & 0 & 0 & 0 \\
3 & -0.6667 & 0 & 0.9547 & 0 & -0.1440 & -0.1920 & 0 & 0 & -0.1440 & 0.1920 \\
4 & 0 & 0 & 0 & 1.0120 & -0.1920 & -0.2560 & 0 & -0.5000 & 0.1920 & -0.2560 \\
5 & 0 & 0 & -0.1440 & -0.1920 & 0.8107 & 0.1920 & -0.6667 & 0 & 0 & 0 \\
6 & 0 & -0.5000 & -0.1920 & -0.2560 & 0.1920 & 0.7560 & 0 & 0 & 0 & 0 \\
7 & 0 & 0 & 0 & 0 & -0.6667 & 0 & 1.3333 & 0 & -0.6667 & 0 \\
8 & 0 & 0 & 0 & -0.5000 & 0 & 0 & 0 & 0.5000 & 0 & 0 \\
9 & 0 & 0 & -0.1440 & 0.1920 & 0 & 0 & -0.6667 & 0 & 0.8107 & -0.1920 \\
10 & 0 & 0 & 0.1920 & -0.2560 & 0 & 0 & 0 & 0 & -0.1920 & 0.2560
\end{array}
\begin{bmatrix} 0 \\ 0 \\ \Delta 2x \\ \Delta 2y \\ 0 \\ \Delta 3y \\ \Delta 4x \\ \Delta 4y \\ 0 \\ 0 \end{bmatrix}
=
\begin{bmatrix} R1x \\ R1y \\ 10 \\ 0 \\ R3x \\ 0 \\ 0 \\ -10 \\ R5x \\ R5y \end{bmatrix}
$$

Rearranging:

$$
1.0e{+}06\ *\
\begin{array}{c|cccccccccc}
 & 3 & 4 & 6 & 7 & 8 & 1 & 2 & 5 & 9 & 10 \\
\hline
3 & 0.9547 & 0 & -0.1920 & 0 & 0 & -0.6667 & 0 & -0.1440 & -0.1440 & 0.1920 \\
4 & 0 & 1.0120 & -0.2560 & 0 & -0.5000 & 0 & 0 & -0.1920 & 0.1920 & -0.2560 \\
6 & -0.1920 & -0.2560 & 0.7560 & 0 & 0 & 0 & -0.5000 & 0.1920 & 0 & 0 \\
7 & 0 & 0 & 0 & 1.3333 & 0 & 0 & 0 & -0.6667 & -0.6667 & 0 \\
8 & 0 & -0.5000 & 0 & 0 & 0.5000 & 0 & 0 & 0 & 0 & 0 \\
1 & -0.6667 & 0 & 0 & 0 & 0 & 0.6667 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & -0.5000 & 0 & 0 & 0 & 0.5000 & 0 & 0 & 0 \\
5 & -0.1440 & -0.1920 & 0.1920 & -0.6667 & 0 & 0 & 0 & 0.8107 & 0 & 0 \\
9 & -0.1440 & 0.1920 & 0 & -0.6667 & 0 & 0 & 0 & 0 & 0.8107 & -0.1920 \\
10 & 0.1920 & -0.2560 & 0 & 0 & 0 & 0 & 0 & 0 & -0.1920 & 0.2560
\end{array}
\begin{bmatrix} \Delta 2x \\ \Delta 2y \\ \Delta 3y \\ \Delta 4x \\ \Delta 4y \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
=
\begin{bmatrix} 10 \\ 0 \\ 0 \\ 0 \\ -10 \\ R1x \\ R1y \\ R3x \\ R5x \\ R5y \end{bmatrix}
$$

In general :

unknown

$$
\begin{bmatrix} \underset{\sim}{K}^G_{ff} & \underset{\sim}{K}^G_{fs} \\ \underset{\sim}{K}^G_{sf} & \underset{\sim}{K}^G_{ss} \end{bmatrix}
\begin{bmatrix} \underline{d}^G_f \\ \underline{d}^G_s \end{bmatrix}
=
\begin{bmatrix} \underline{f}^G_f \\ \underline{f}^G_s \end{bmatrix}
$$

given

$\Rightarrow$ Solve $\boxed{[\underset{\sim}{K}^G_{ff}]\{\underline{d}^G_f\} = \{\underline{f}^G_f\} - [\underset{\sim}{K}^G_{fs}]\{\underline{d}^G_s\}}$ for $\underline{\underline{d}^G_f}$

and

$$\underline{f}^G_s = [\underset{\sim}{K}^G_{sf}]\{\underline{d}^G_f\} + [\underset{\sim}{K}^G_{ss}]\{\underline{d}^G_s\}$$ (support reactions)

Displacement vector:

u =

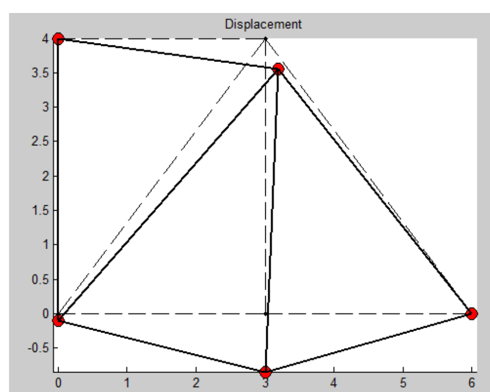
1	0
2	0
3	0.009454976303318
4	-0.022066795023697
5	0
6	-0.005071090047393
7	0
8	-0.042066795023697
9	0
10	0

Reactions (Force vector)

f =

1.0e+04 *

1	-0.630331753554502
2	0.253554502369668
3	1.000000000000000
4	0
5	0.190165876777251
6	0
7	0
8	-1.000000000000000
9	-0.559834123222749
10	0.746445497630332



MATLAB Code for 2D Truss Analysis using the Stiffness Method

```
% 2D Truss code

clear all; clc; close all; % clear all the existing variables (new start)

% Obtain the input file name from the user & Read Input
inpfilename = uigetfile('*.txt','Select the input file');

[nodes, elems, C, A, bcs, loads] = gettrussdata2D(inpfilename);
Nel = size(elems,1);
Nnodes = size(nodes,1);

% Decide degrees of freedom + Initialize Matrices
alldofs = 1:2*Nnodes;
K = zeros(2*Nnodes);
u = zeros(2*Nnodes,1);
f = zeros(2*Nnodes,1);
% Note: Degrees of Freedom correspoding to node "i"
% are [2*(i-1)+1   2*(i-1)+2]

% Boundary conditions
dofspec = [];
for ii = 1:size(bcs,1)
    thisdof = 2*(bcs(ii,1)-1)+bcs(ii,2);
    dofspec = [dofspec thisdof];
    u(thisdof) = bcs(ii,3);
end
doffree = alldofs;
doffree(dofspec) = []; % Delete specified dofs from All dofs

% Nodal Loads
for ii = 1:size(loads,1)
    f(2*(loads(ii,1)-1)+loads(ii,2)) = loads(ii,3);
end

% Initialize the global stiffness matrix
for iel = 1:Nel
    elnodes = elems( iel, 1:2);
    nodexy = nodes(elnodes, :);

    % Get the element stiffness matrix for the current element
    [Kel] = TrussElement2D(nodexy, C(iel), A(iel));

    % Assemble the element stiffness matrix into the global stiffness matrix K
    eldofs = 2*(elnodes(1)-1)+1:2*(elnodes(1)-1)+2;
    eldofs = [eldofs 2*(elnodes(2)-1)+1:2*(elnodes(2)-1)+2];
    K(eldofs,eldofs) = K(eldofs,eldofs) + Kel;
end

% Solve
u(doffree) = K(doffree,doffree)\(f(doffree)-K(doffree,dofspec)*u(dofspec));
f(dofspec) = K(dofspec,:)*u;
format long
disp(['Displacement vector:']); u
disp(['Reactions (Force vector)']); f
```

Input File

```
Nodes: (x, y)
0.0     4.0
3.0     4.0
0.0     0.0
3.0     0.0
6.0     0.0
Elements: (Node1 Node2), E, A,
1 2 2e11 1e-5
1 3 2e11 1e-5
3 2 2e11 1e-5
2 4 2e11 1e-5
2 5 2e11 1e-5
3 4 2e11 1e-5
4 5 2e11 1e-5
BCs (Node_number dof specified_disp)
1 1 0
1 2 0
3 1 0
5 1 0
5 2 0
Nodal loads (Node_number dof
2 1 1e4
4 2 -1e4
```

MATLAB Code for 2D Truss Analysis using the Stiffness Method (Continued)

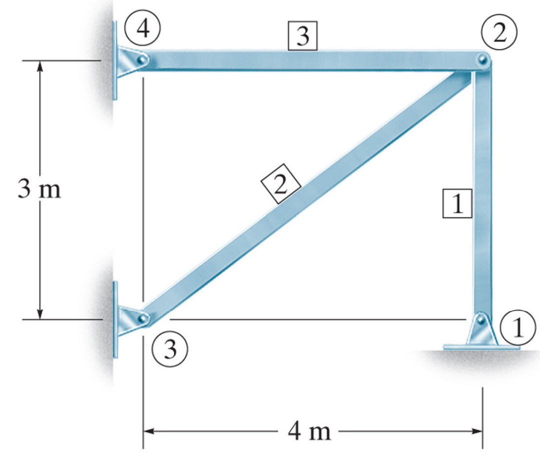
```
% plot old shape
figure(1); hold on;
plot(nodes(:,1),nodes(:,2),'k.')
hold on; axis equal;
for iel = 1:Nel
    elnodes = elems(iel, 1:2);
    nodexy = nodes(elnodes, :);
    plot(nodexy(:,1),nodexy(:,2),'k--')
end
% plot new shape
Magnification = 20;
nodesnew = nodes + Magnification*reshape(u,2,Nnodes)';
plot(nodesnew(:,1),nodesnew(:,2),'o', ...
    'MarkerEdgeColor','k', 'MarkerFaceColor','r','MarkerSize',10)
hold on; axis equal;
for iel = 1:Nel
    elnodes = elems(iel, 1:2);
    nodexy = nodesnew(elnodes, :);
    plot(nodexy(:,1),nodexy(:,2),'k-','LineWidth',2)
end
title('Displacement');
```

Calculation of Local and Global Element Stiffness Matrices

```
function [Kel] = TrussElement2D(nodexy, C, A)
% This function must return a 4x4 element stiffness matrix: [Kel]
% This matrix must be in the GLOBAL Coordinates
% Input:
% nodexy : [ x1 y1 ;
%            x2 y2 ]
% C : Youngs modulus
% A : Area of cross-section

E1 = [ (nodexy(2,1)-nodexy(1,1)) ...
       (nodexy(2,2)-nodexy(1,2)) ];
le = norm(E1);
E1 = E1/le;
E2 = [-E1(2) E1(1)];

Kel_LOC = zeros(4);
Kel_LOC([1 3],[1 3]) = C*A/le*[1 -1; -1 1];

Qrot = [E1; E2]; % Transforms global to element d_E = Q d_G
Tmatrix = [Qrot zeros(2); zeros(2) Qrot];

Kel = Tmatrix'*Kel_LOC*Tmatrix;
```

Example

Support at node 1 settles down by 25mm.
Determine the force in member 2.
AE = 8×10^6 N



(a)

Figure: 14_11aEX05


```
K1 =
   1.0e+06 *
         0         0         0         0
         0    2.6667         0   -2.6667
         0         0         0         0
         0   -2.6667         0    2.6667
K2 =
     1024000      768000    -1024000     -768000
      768000      576000     -768000     -576000
    -1024000     -768000     1024000      768000
     -768000     -576000      768000      576000
K3 =
     2000000           0    -2000000           0
           0           0           0           0
    -2000000           0     2000000           0
           0           0           0           0
```

Kglobal = 1.0e+06 *

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.6667 & 0 & -2.6667 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.0240 & 0.7680 & -1.0240 & -0.7680 & -2.0000 & 0 \\ 0 & -2.6667 & 0.7680 & 3.2427 & -0.7680 & -0.5760 & 0 & 0 \\ 0 & 0 & -1.0240 & -0.7680 & 1.0240 & 0.7680 & 0 & 0 \\ 0 & 0 & -0.7680 & -0.5760 & 0.7680 & 0.5760 & 0 & 0 \\ 0 & 0 & -2.0000 & 0 & 0 & 0 & 2.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Solution:
Displacements:

```
u =
         0
   -0.0250
    0.0056
   -0.0219
         0
         0
         0
         0
```

Reactions:

```
f =
   1.0e+04 *
         0
   -0.8333
         0
         0
    1.1111
    0.8333
   -1.1111
         0
```

Displacement of member 2

```
>> u2=u([5,6,3,4])
u2 =
         0
         0
    0.0056
   -0.0219
```

Force in Member 2

```
>> f2 = K2 * u2
f2 =
   1.0e+04 *
    1.1111
    0.8333
   -1.1111
   -0.8333
```

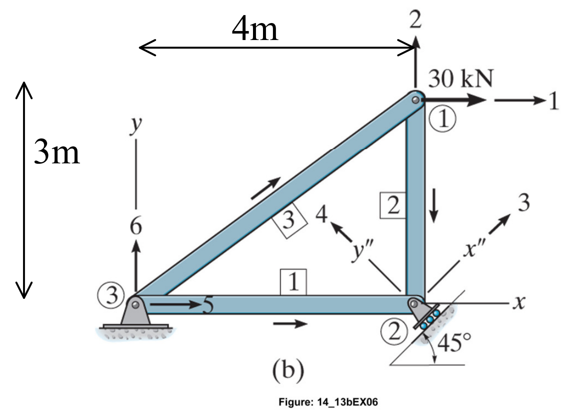
```
>> T2'*f2
ans =
   1.0e+04 *
    1.3889
    0.0000
   -1.3889
   -0.0000
```

Inclined Support Conditions

Sometimes, the support conditions are not oriented along global x-y axis.

In these cases, one must transform specific components of the global equilibrium equations to match the orientation of the inclined supports so that the boundary conditions can be enforced correctly.

Example



(b)

Figure: 14_13bEX06

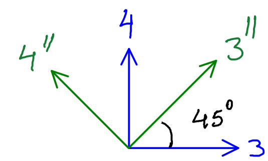
Original global system:

$$\left[\begin{array}{c|c} \underset{\sim}{K}_{ff}^{G} & \underset{\sim}{K}_{fs}^{G} \\ \hline \underset{\sim}{K}_{sf}^{G} & \underset{\sim}{K}_{ss}^{G} \end{array}\right] \left[\begin{array}{c} \underline{d}_f^{G} \\ \hline \underline{d}_s^{G} \end{array}\right] = \left[\begin{array}{c} \underline{f}_f^{G} \\ \hline \underline{f}_s^{G} \end{array}\right]$$

$$\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \left[\quad \underset{\sim}{K}^{G}_{6\times 6} \quad \right] \left[\begin{array}{c} d_1 \\ d_2 \\ \boxed{\begin{array}{c} d_3 \\ d_4 \end{array}} \\ d_5 \\ d_6 \end{array}\right] = \left[\underset{6\times 1}{f^{G}} \right]$$

Note:

$$\begin{Bmatrix} d_3'' \\ d_4'' \end{Bmatrix} = \underbrace{\begin{bmatrix} c & s \\ -s & c \end{bmatrix}}_{\underset{\sim}{T}} \begin{Bmatrix} d_3 \\ d_4 \end{Bmatrix}$$



Degrees of freedom 3 and 4 need to be rotated to 3" and 4"

$$\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \left[\quad \underset{\sim}{K}^{G}_{6\times 6} \quad \right] \underbrace{\begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & c & -s & & \\ & & s & c & & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix}}_{\underset{\sim}{T}_G^{T}} \underbrace{\begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & c & s & & \\ & & -s & c & & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix}}_{\underset{\sim}{T}_G} \left[\begin{array}{c} d_1 \\ d_2 \\ \boxed{\begin{array}{c} d_3 \\ d_4 \end{array}} \\ d_5 \\ d_6 \end{array}\right] = \left[\underset{6\times 1}{f^{G}} \right]$$

Modify the original global system: $\underset{\sim}{T}_G \left(\underset{\sim}{K}^{g} \, \underset{\sim}{T}_G^{T} \underset{\sim}{T}_G \, \underline{d}^{g} = \underline{f}^{g} \right)$

$$\underbrace{\left(\underset{\sim}{T} \, \underset{\sim}{K}^{G} \, \underset{\sim}{T}^{T}\right)}_{\left[\underset{\sim}{K}^{G'}\right]} \underbrace{\left(\underset{\sim}{T} \, \underline{d}^{G}\right)}_{\left\{\underline{d}^{G'}\right\}} = \underbrace{\left(\underset{\sim}{T} \, \underline{f}^{G}\right)}_{\left\{\underline{f}^{G'}\right\}}$$

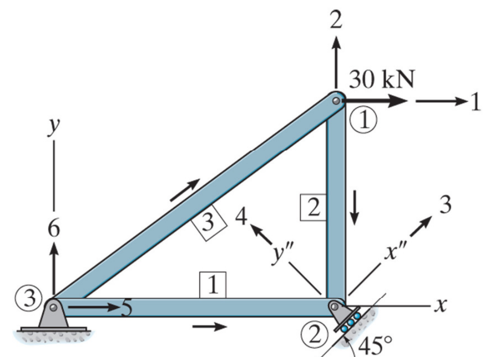
Now Enforce BCs & solve.

$$\left[\begin{array}{c|c} \underset{\sim}{K}_{ff}^{G'} & \underset{\sim}{K}_{fs}^{G'} \\ \hline \underset{\sim}{K}_{sf}^{G'} & \underset{\sim}{K}_{ss}^{G'} \end{array}\right] \left[\begin{array}{c} \underline{d}_f^{G'} \\ \hline \underline{d}_s^{G'} \end{array}\right] = \left[\begin{array}{c} \underline{f}_f^{G'} \\ \hline \underline{f}_s^{G'} \end{array}\right]$$

Example

Find displacements and reactions.

Assume EA = 1



(b)

Figure: 14_13bEX06

K1 =

0.2500	0	-0.2500	0
0	0	0	0
-0.2500	0	0.2500	0
0	0	0	0

K2 =

0	0	0	0
0	0.3333	0	-0.3333
0	0	0	0
0	-0.3333	0	0.3333

K3 =

0.1280	0.0960	-0.1280	-0.0960
0.0960	0.0720	-0.0960	-0.0720
-0.1280	-0.0960	0.1280	0.0960
-0.0960	-0.0720	0.0960	0.0720

TG =

1.0000	0	0	0	0	0
0	1.0000	0	0	0	0
0	0	0.7071	0.7071	0	0
0	0	-0.7071	0.7071	0	0
0	0	0	0	1.0000	0
0	0	0	0	0	1.0000

K^G =

0.1280	0.0960	0	0	-0.1280	-0.0960
0.0960	0.4053	0	-0.3333	-0.0960	-0.0720
0	0	0.2500	0	-0.2500	0
0	-0.3333	0	0.3333	0	0
-0.1280	-0.0960	-0.2500	0	0.3780	0.0960
-0.0960	-0.0720	0	0	0.0960	0.0720

$K^{G'}$ =

0.1280	0.0960	0	0	-0.1280	-0.0960
0.0960	0.4053	-0.2357	-0.2357	-0.0960	-0.0720
0	-0.2357	0.2917	0.0417	-0.1768	0
0	-0.2357	0.0417	0.2917	0.1768	0
-0.1280	-0.0960	-0.1768	0.1768	0.3780	0.0960
-0.0960	-0.0720	0	0	0.0960	0.0720

Solution:

u1 =

1.0e+05 *

3.5250
-1.5750
-1.2728
0
0
0

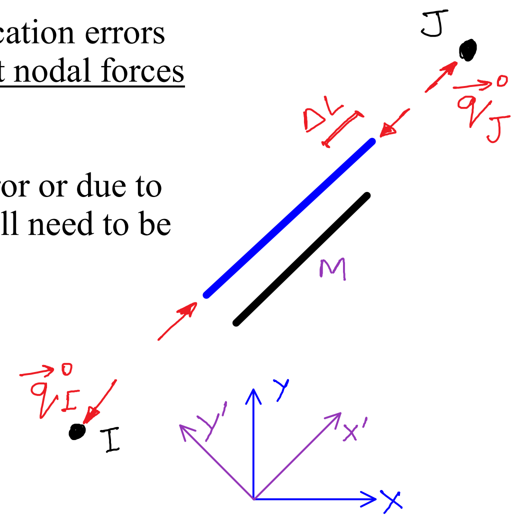
f1 =

1.0e+04 *

3.0000
0
0
3.1820
-0.7500
-2.2500

Effect of Temperature Changes and Fabrication Errors

Changes in lengths of truss members due to temperature or fabrication errors can also be accommodated in the analysis by applying equivalent nodal forces that would result from these changes.



If a member has change in length ΔL (either due to fabrication error or due to temperature $\Delta L = \alpha \, \Delta T \, L$) then the equivalent nodal forces that will need to be applied to the truss will be:

$$\begin{Bmatrix} \underline{q}_I^{o'} \\ \underline{q}_J^{o'} \end{Bmatrix} = AE \frac{\Delta L}{L} \begin{Bmatrix} -1 \\ 0 \\ 1 \\ 0 \end{Bmatrix}$$

$$\Rightarrow \begin{Bmatrix} \underline{q}_I^{o} \\ \underline{q}_J^{o} \end{Bmatrix} = \begin{bmatrix} \underset{\sim}{T}_M & 0 \\ 0 & \underset{\sim}{T}_M \end{bmatrix} \begin{Bmatrix} \underline{q}_I^{o'} \\ \underline{q}_J^{o'} \end{Bmatrix}$$

$$\vec{q}^{\,o} \longrightarrow \begin{Bmatrix} \underline{q}_I^{o'} \\ \underline{q}_J^{o'} \end{Bmatrix} \longrightarrow \begin{Bmatrix} \underline{q}_I^{o} \\ \underline{q}_J^{o} \end{Bmatrix}$$

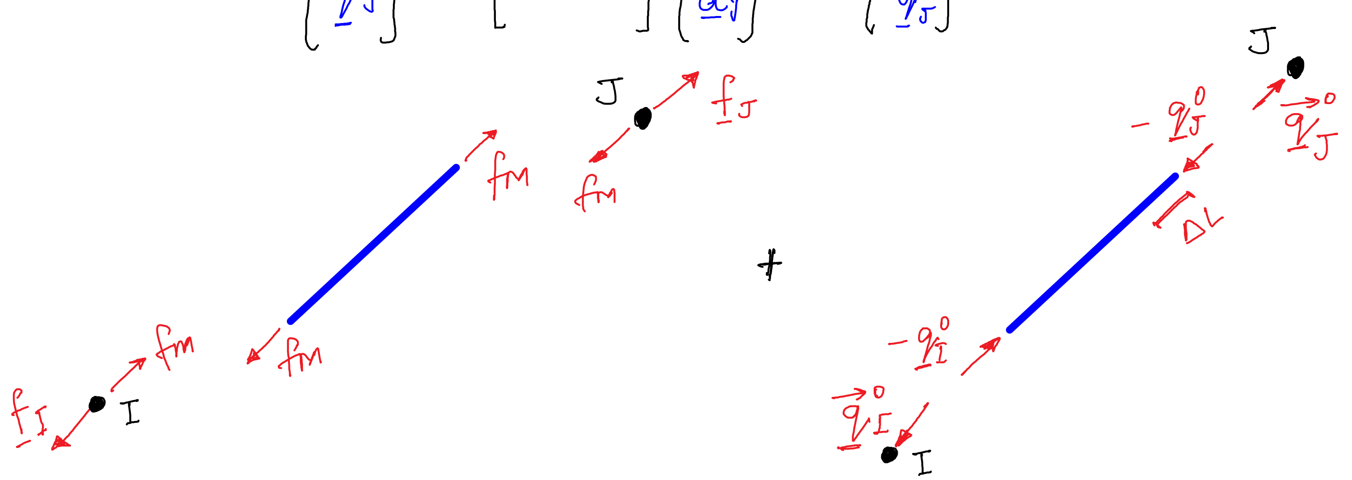
$$\begin{Bmatrix} \underline{q}_I^{o} \\ \underline{q}_J^{o} \end{Bmatrix} = AE \frac{\Delta L}{L} \begin{bmatrix} \underset{\sim}{T}_M & 0 \\ 0 & \underset{\sim}{T}_M \end{bmatrix} \begin{Bmatrix} -1 \\ 0 \\ 1 \\ 0 \end{Bmatrix}$$

This force would need to be added to the dofs of I and J as external loads.

$$\left[\begin{array}{c|c} \underset{\sim}{K}_{ff}^G & \underset{\sim}{K}_{fs}^G \\ \hline \underset{\sim}{K}_{sf}^G & \underset{\sim}{K}_{ss}^G \end{array}\right] \left[\begin{array}{c} \underline{d}_f^G \\ \hline \underline{d}_s^G \end{array}\right] = \left[\begin{array}{c} \underline{f}_f^G \\ \hline \underline{f}_s^G \end{array}\right] + \begin{bmatrix} \underline{q}_f^o \\ \underline{q}_s^o \end{bmatrix}$$

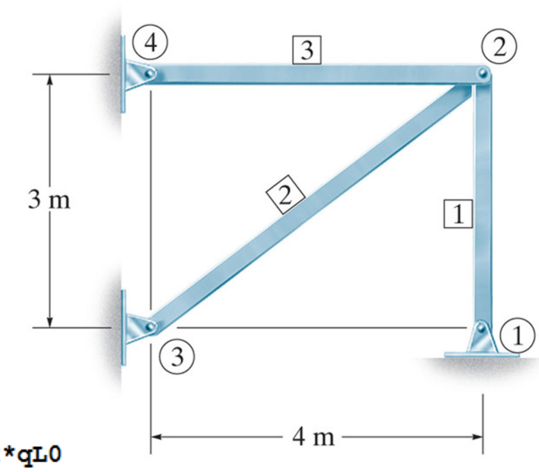
Note: Once displacements have been found the internal forces in member M can be found by:

$$\begin{Bmatrix} \underline{q}_I \\ \underline{q}_J \end{Bmatrix} = \begin{bmatrix} \underset{\sim}{K}_M \end{bmatrix} \begin{Bmatrix} \underline{d}_I \\ \underline{d}_J \end{Bmatrix} + \begin{Bmatrix} -\underline{q}_I^o \\ -\underline{q}_J^o \end{Bmatrix}$$



Example

Member 2 is too short by 0.01 m.
Determine the force in member 2.
AE = $8\text{x}10^6$ N




```
>> qL0 = 8e6*(-0.01)/5*[-1 0 1 0]'
qL0 =
       16000
           0
      -16000
           0
```

```
T2 =
    0.8000   -0.6000         0         0
    0.6000    0.8000         0         0
         0         0    0.8000   -0.6000
         0         0    0.6000    0.8000
```

```
>> q0 = T2*qL0
q0 =
       12800
        9600
      -12800
       -9600
```

Kglobal = 1.0e+06 *

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.6667 & 0 & -2.6667 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.0240 & 0.7680 & -1.0240 & -0.7680 & -2.0000 & 0 \\ 0 & -2.6667 & 0.7680 & 3.2427 & -0.7680 & -0.5760 & 0 & 0 \\ 0 & 0 & -1.0240 & -0.7680 & 1.0240 & 0.7680 & 0 & 0 \\ 0 & 0 & -0.7680 & -0.5760 & 0.7680 & 0.5760 & 0 & 0 \\ 0 & 0 & -2.0000 & 0 & 0 & 0 & 2.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Solution

```
u =
          0
          0
    -0.0037
    -0.0021
          0
          0
          0
          0
```

```
f =
   1.0e+04 *
         0
    0.5556
   -1.2800
   -0.9600
    0.5393
    0.4044
    0.7407
         0
```

Force in member 2:

```
>> u2 = u([5 6 3 4])
u2 =
          0
          0
    -0.0037
    -0.0021
```

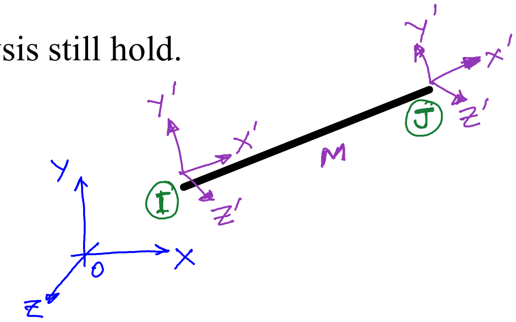
```
>> f2 = K2*u2-q0
f2 =
   1.0e+03 *
   -7.4074
   -5.5556
    7.4074
    5.5556
```

```
>> T2'*f2
ans =
   1.0e+03 *
   -9.2593
         0
    9.2593
         0
```

Space (3D) Truss Analysis

For space (3D) trusses, all the same concepts of 2D truss analysis still hold.
The main differences are:

- 3 dofs per node
- Transformation matrix becomes 3x3

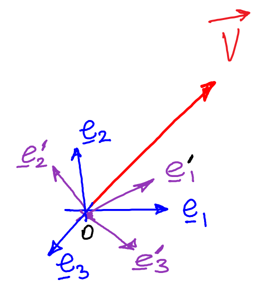


Coordinate Transformation

$$\vec{V} = \sum_i v_i \underset{\sim}{e}_i = v_1 \underset{\sim}{e}_1 + v_2 \underset{\sim}{e}_2 + v_3 \underset{\sim}{e}_3 \rightarrow \begin{Bmatrix} v_1 \\ v_2 \\ v_3 \end{Bmatrix}_{(\underset{\sim}{e}_1, \underset{\sim}{e}_2, \underset{\sim}{e}_3)}$$

Also

$$\vec{V} = \sum_i v_i' \underset{\sim}{e}_i' = v_1' \underset{\sim}{e}_1' + v_2' \underset{\sim}{e}_2' + v_3' \underset{\sim}{e}_3' \rightarrow \begin{Bmatrix} v_1' \\ v_2' \\ v_3' \end{Bmatrix}_{(\underset{\sim}{e}_1', \underset{\sim}{e}_2', \underset{\sim}{e}_3')}$$



To find v_i' :

$$\vec{V} \cdot \underset{\sim}{e}_j' = v_j' = \sum_{i=1}^{3} v_i' \underbrace{(\underset{\sim}{e}_i' \cdot \underset{\sim}{e}_j')}_{\delta_{ij}} = \sum_{i=1}^{3} v_i \underbrace{(\underset{\sim}{e}_i \cdot \underset{\sim}{e}_j')}_{Q_{ji} = \cos(\theta_{ij'})}$$

$(j = 1, 2, 3)$

$$\delta_{ij} = \begin{cases} 0 & : i \neq j \\ 1 & : i = j \end{cases}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Thus

$$\boxed{v_j' = Q_{ji} \; v_i}$$

In matrix form

$$\begin{Bmatrix} v_1' \\ v_2' \\ v_3' \end{Bmatrix} = \underbrace{\begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix}}_{\underset{\sim}{T}_M^T} \begin{Bmatrix} v_1 \\ v_2 \\ v_3 \end{Bmatrix}$$

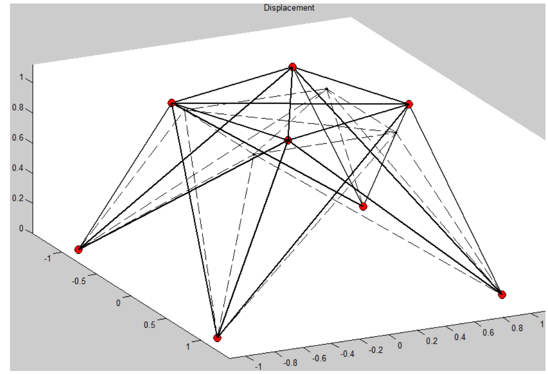
Element Stiffness Matrix :
(in local co-ordinates)

$$\underset{\sim}{K}^{M'} = \frac{EA}{L} \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

(in global coordinates):

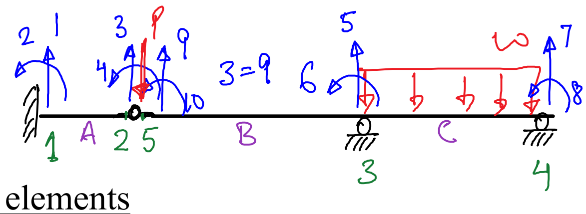
$$\left[\underset{\sim}{K}_M\right] = \begin{bmatrix} \underset{\sim}{T}_M & 0 \\ 0 & \underset{\sim}{T}_M \end{bmatrix} \left[\underset{\sim}{K}_M'\right] \begin{bmatrix} \underset{\sim}{T}_M^T & 0 \\ 0 & \underset{\sim}{T}_M^T \end{bmatrix}$$

Example




```
Nodes: (x, y, z)
-1          -1          0
 1          -1          0
 1           1          0
-1           1          0
-0.5        -0.5         1
 0.5        -0.5         1
 0.5         0.5         1
-0.5         0.5         1
Elements: (Node1 Node2), Orientation (E2x, E2y, E2z), C, A,
1 5  -0.57735026918963  -0.57735026918963   0.57735026918963  1 1
1 6  -0.15430334996209  -0.77151674981046   0.61721339984837  1 1
5 2   0.15430334996209  -0.77151674981046   0.61721339984837  1 1
2 6   0.57735026918963  -0.57735026918963   0.57735026918963  1 1
2 7   0.77151674981046  -0.15430334996209   0.61721339984837  1 1
6 3   0.77151674981046   0.15430334996209   0.61721339984837  1 1
3 7   0.57735026918963   0.57735026918963   0.57735026918963  1 1
3 8   0.15430334996209   0.77151674981046   0.61721339984837  1 1
7 4  -0.15430334996209   0.77151674981046   0.61721339984837  1 1
4 8  -0.57735026918963   0.57735026918963   0.57735026918963  1 1
4 5  -0.77151674981046   0.15430334996209   0.61721339984837  1 1
1 8  -0.77151674981046  -0.15430334996209   0.61721339984837  1 1
5 7   0                  0                  1                 1 1
6 8   0                  0                  1                 1 1
5 6   0                 -0.44721359549996   0.89442719099992  1 1
6 7   0.44721359549996   0                  0.89442719099992  1 1
7 8   0                  0.44721359549996   0.89442719099992  1 1
8 5  -0.44721359549996   0                  0.89442719099992  1 1
BCs (Node_number specified_dx specified_dy specified_dz)
1 0 0 0
2 0 0 0
3 0 0 0
4 0 0 0
Nodal loads (Node_number fx fy fz)
5   0.1      -0.1     0.1
6   0.1       0.1     0.1
7  -0.1       0.1     0.1
8  -0.1      -0.1     0.1
```

Stiffness method for Beams



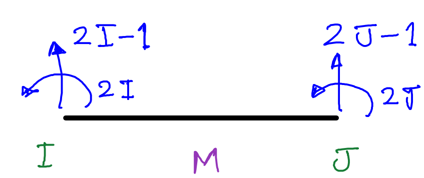
The overall methodology of the stiffness methods is still the same for problems involving beams:

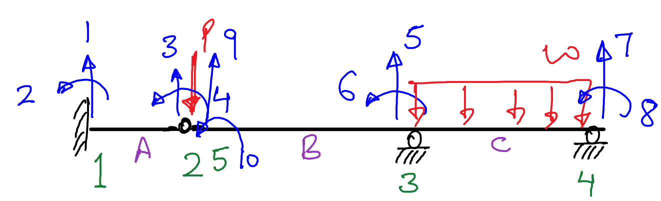
1. Define the geometry of the problem in terms of nodes and elements
2. Set up the degrees of freedom: transverse displacements and rotations at nodes
3. Define the loading and boundary conditions as externally applied forces and moments, and degrees of freedom that are fixed / specified.
4. Set up element force-displacement relations $\boxed{\boldsymbol{q}_M = \boldsymbol{K}_M \,.\, \boldsymbol{d}_M}$ (local and global coordinate systems are the same)
5. Assemble forces and moments from all elements in terms of unknown global displacements and rotations

$$\left[\begin{array}{c|c} \underset{\sim}{K}^G_{ff} & \underset{\sim}{K}^G_{fs} \\ \hline \underset{\sim}{K}^G_{sf} & \underset{\sim}{K}^G_{ss} \end{array}\right] \left[\begin{array}{c} \underline{d}^G_f \\ \hline \underline{d}^G_s \end{array}\right] = \left[\begin{array}{c} \underline{f}^G_f \\ \hline \underline{f}^G_s \end{array}\right]$$

Solve by partitioning the free and specified degrees of freedom as usual.

Nodes Elements and Degrees of Freedom

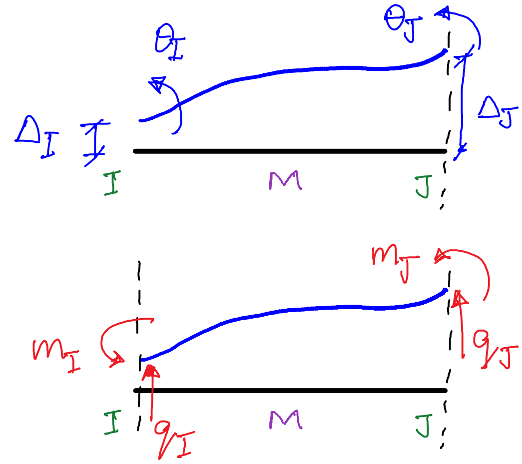


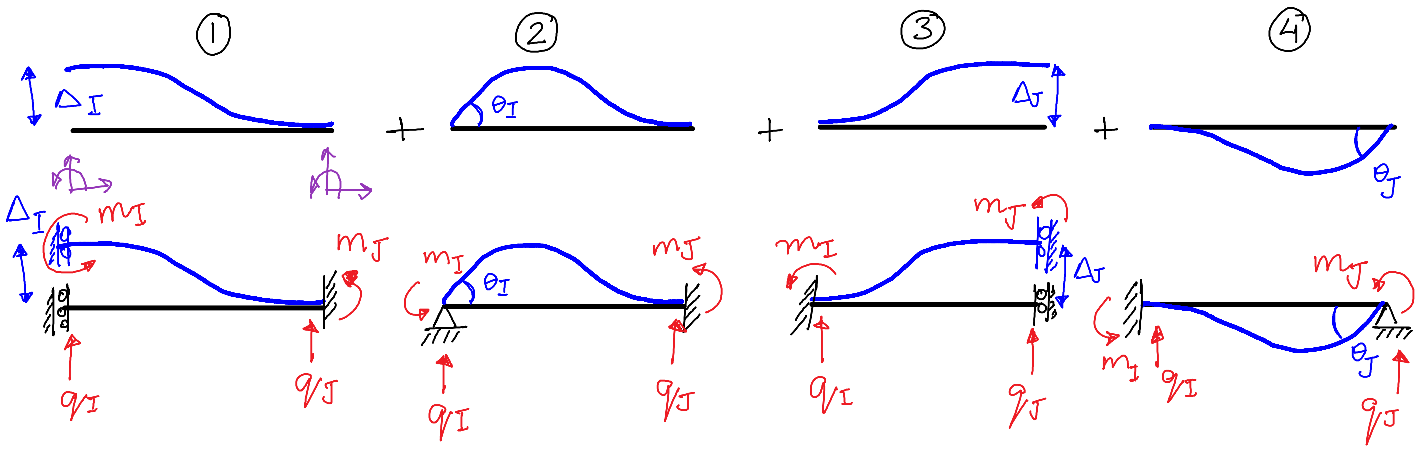


Element force-displacement relationship

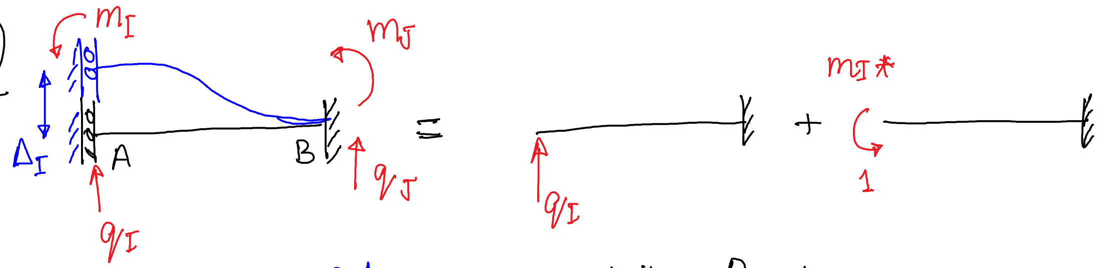
Forces and Moments developed in a beam element :

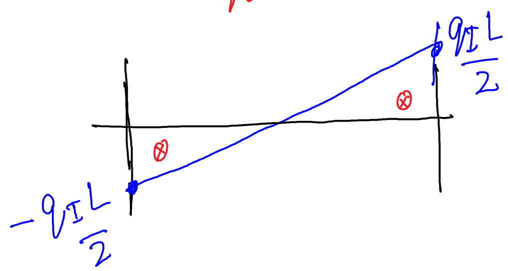
$$\begin{Bmatrix} q_I \\ m_I \\ q_J \\ m_J \end{Bmatrix}_{4\times1} = \begin{bmatrix} * & & \\ * & & \underset{\sim}{K}^M \\ * & & \\ * & & \end{bmatrix}_{4\times4} \begin{Bmatrix} \Delta_I \\ \theta_I \\ \Delta_J \\ \theta_J \end{Bmatrix}_{4\times1}$$



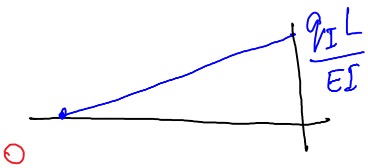


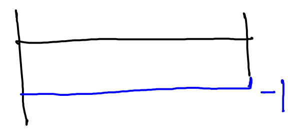
Case ①





Compatibility: $\underbrace{\theta_I + m_I^* \, \alpha_{AA} = 0}$





$$\Delta_I = -\frac{1}{2}\left(\frac{q_I L}{2EI}\right)\left(\frac{L}{2}\right)\times\left(\frac{L}{2}\times\frac{1}{3}\right) + \frac{1}{2}\left(\frac{q_I L}{2EI}\right)\left(\frac{L}{2}\right)\times\left(\frac{L}{2}+\frac{2}{3}\frac{L}{2}\right)$$

$$= \frac{1}{2}\,\frac{q_I L}{2EI}\cdot\frac{L}{2}\cdot\frac{L}{2}\cdot\left(1+\frac{1}{3}\right)$$

$$\boxed{\Delta_I = \frac{q_I L^3}{12EI}}$$

$$\overset{0}{\theta_J} - \theta_I = \frac{1}{2}\,\frac{q_I L}{EI}\times L \quad\Rightarrow\quad \theta_I = \frac{q_I L^2}{2EI}$$

$$\alpha_{AA} = \frac{-1\times L}{EI}$$

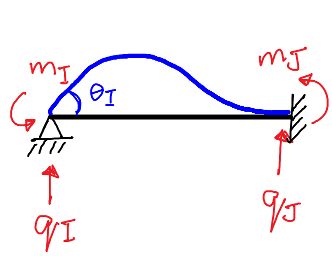
Compatibility:

$$m_I = \frac{-\theta_I}{\alpha_{AA}} = \frac{q_I L^{\cancel{2}}}{2\cancel{EI}}\cdot\frac{\cancel{EI}}{\cancel{L}}$$

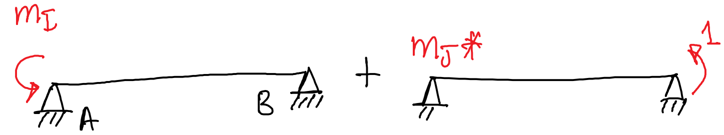
$$\boxed{m_I = \frac{q_I L}{2}}$$

$$\left.\begin{aligned}
q_I &= \frac{12EI}{L^3}\Delta_I \\
m_I &= \frac{6EI}{L^2}\Delta_I \\
q_J &= -\frac{12EI}{L^3}\Delta_I \\
m_J &= q_I L - m_I = \frac{6EI}{L^2}\Delta_I
\end{aligned}\right\} \Rightarrow
\begin{Bmatrix} q_I \\ m_I \\ q_J \\ m_J \end{Bmatrix} =
\begin{bmatrix} 12EI/L^3 \\ 6EI/L^2 \\ -12EI/L^3 \\ 6EI/L^2 \end{bmatrix}
\begin{Bmatrix} \Delta_I \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

Case ②



=



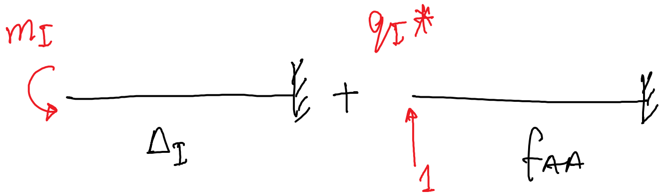
Comp: $\underbrace{\theta_J + m_J^* \, \alpha_{BB} = 0}$

$$q_J = -q_I = \boxed{\frac{-3m_I}{2L}}$$

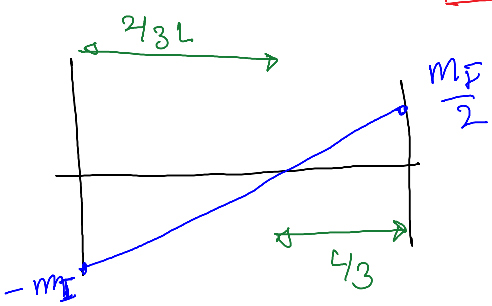
$$m_J = q_I \times L - m_I$$

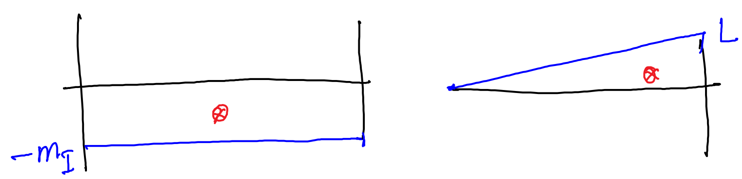
$$m_J = \frac{3m_I}{2} - m_I = \boxed{\frac{m_I}{2}}$$

OR =



Compatibility: $\underbrace{\Delta_I + q_I \times f_{AA} = 0}$





$$\Delta_I = \left(\frac{-m_I}{EI} \times L\right) \times \frac{L}{2} \qquad f_{AA} = \left(\frac{1}{\cancel{2}} \times \left(\frac{L}{EI}\right) \times L\right) \times \frac{\cancel{2}}{3} L$$

$$-\theta_I = \left(-\frac{1}{\cancel{2}} \frac{m_I}{EI} \times \frac{\cancel{2}}{3} L\right) + \left(\frac{1}{2} \frac{m_I}{2EI} \cdot \frac{L}{3}\right)$$

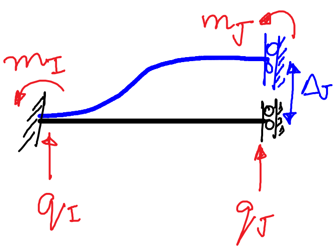
$$q_I = \frac{-\Delta_I}{f_{AA}} = \frac{m_I L^{\cancel{2}}}{2\cancel{EI}} \cdot \frac{3\cancel{EI}}{L^{\cancel{2}}} = \boxed{\frac{3m_I}{2L}}$$

$$\theta_I = \frac{m_I L}{3EI} - \frac{m_I L}{12EI} = \boxed{\frac{1}{4}\frac{L}{EI} m_I}$$

$$\left.\begin{aligned} q_I &= \frac{6EI}{L^2}\theta_I \\ m_I &= \frac{4EI}{L}\theta_I \\ q_J &= \frac{-6EI}{L^2}\theta_I \\ m_J &= \frac{2EI}{L}\theta_I \end{aligned}\right\} \Rightarrow \begin{Bmatrix} q_I \\ m_I \\ q_J \\ m_J \end{Bmatrix} = \begin{bmatrix} \frac{6EI}{L^2} \\ \frac{4EI}{L} \\ \frac{-6EI}{L^2} \\ \frac{2EI}{L} \end{bmatrix} \begin{Bmatrix} 0 \\ \theta_I \\ 0 \\ 0 \end{Bmatrix}$$

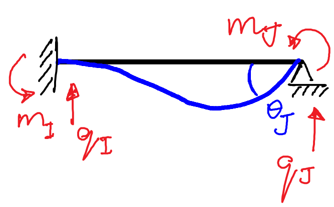
Using information from cases 1 and 2 above:

Case ③



$$\Rightarrow \quad \begin{Bmatrix} q_I \\ m_I \\ q_J \\ m_J \end{Bmatrix} = \begin{bmatrix} -12EI/L^3 \\ -6EI/L^2 \\ 12EI/L^3 \\ -6EI/L^2 \end{bmatrix} \begin{Bmatrix} 0 \\ 0 \\ \Delta_J \\ 0 \end{Bmatrix}$$

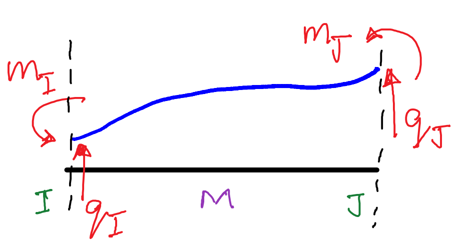
Case ④



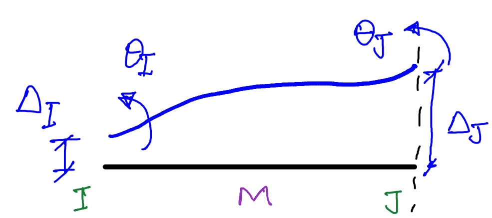
$$\Rightarrow \quad \begin{Bmatrix} q_I \\ m_I \\ q_J \\ m_J \end{Bmatrix} = \begin{bmatrix} \frac{6EI}{L^2} \\ \frac{2EI}{L} \\ \frac{-6EI}{L^2} \\ \frac{4EI}{L} \end{bmatrix} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ \theta_J \end{Bmatrix}$$

Combine Cases ① – ④:

$$\begin{Bmatrix} q_I \\ m_I \\ q_J \\ m_J \end{Bmatrix} = \begin{bmatrix} 12EI/L^3 & \frac{6EI}{L^2} & -12EI/L^3 & \frac{6EI}{L^2} \\ 6EI/L^2 & \frac{4EI}{L} & -6EI/L^2 & \frac{2EI}{L} \\ -12EI/L^3 & \frac{-6EI}{L^2} & 12EI/L^3 & \frac{-6EI}{L^2} \\ 6EI/L^2 & \frac{2EI}{L} & -6EI/L^2 & \frac{4EI}{L} \end{bmatrix} \begin{Bmatrix} \Delta_I \\ \theta_I \\ \Delta_J \\ \theta_J \end{Bmatrix}$$

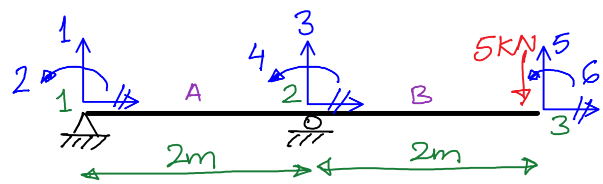


$$\boxed{\underline{q}_M = \underset{\sim}{K}_M \, \underline{d}_M}$$



Sample MATLAB code

```
% Main code for solving 2D Beam problems using Stiffness method

clear all; clc; close all; % clear all the existing variables (new start)

% Obtain the input file name from the user & Read Input
inpfilename = uigetfile('*.txt','Select the input file');

[nodes, elems, E, A, I, bcs, loads] = getbeamdata2D(inpfilename);
Nel = size(elems,1);
Nnodes = size(nodes,1);

% Decide degrees of freedom + Initialize Matrices
alldofs = 1:2*Nnodes;
K = zeros(2*Nnodes);
u = zeros(2*Nnodes,1);
f = zeros(2*Nnodes,1);
% Note: Degrees of Freedom correspoding to node "i"
% are [2*(i-1)+1   2*(i-1)+2]
```




```
Nodes: (x, y)
0.0    0.0
2.0    0.0
4.0    0.0
Elements: (Node1 Node2), E, A, I,
1 2 2e11  1e-2  5e-6
2 3 2e11  1e-2  5e-6
BCs (Node_number dof specified_disp)
1 1 0
2 1 0
Nodal loads (Node_number dof specified load)
3 1 -5000
```

```
% Boundary conditions
dofspec = [];
for ii = 1:size(bcs,1)
    thisdof = 2*(bcs(ii,1)-1)+bcs(ii,2);
    dofspec = [dofspec thisdof];
    u(thisdof) = bcs(ii,3);
end
doffree = alldofs;
doffree(dofspec) = []; % Delete specified dofs from All

% Nodal Loads
for ii = 1:size(loads,1)
    f(2*(loads(ii,1)-1)+loads(ii,2)) = loads(ii,3);
end
% Initialize the global stiffness matrix
for iel = 1:Nel
    elnodes = elems( iel, 1:2);
    nodexy = nodes(elnodes, :);

    % Get the element stiffness matrix for the current element
    [Kel] = BeamElement2DBE(nodexy, E(iel), A(iel), I(iel));

    % Assemble the element stiffness matrix into the global stiffness matrix K
    eldofs = 2*(elnodes(1)-1)+1:2*elnodes(1);
    eldofs = [eldofs 2*(elnodes(2)-1)+1:2*elnodes(2)];
    K(eldofs,eldofs) = K(eldofs,eldofs) + Kel;
end

% Solve
u(doffree) = K(doffree,doffree)\(f(doffree)-K(doffree,dofspec)*u(dofspec));
f(dofspec) = K(dofspec,:)*u;
% format long
disp(['Displacement and Rotations :']); u
disp(['Reactions (Forces and Moments)']); f
```

Plotting

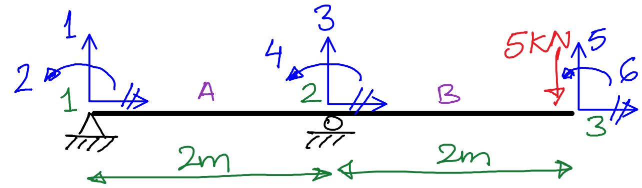
```
% plot old shape
figure(1); hold on;
plot(nodes(:,1),nodes(:,2),'k.')
hold on; axis equal;
for iel = 1:Nel
    elnodes = elems(iel, 1:2);
    nodexy = nodes(elnodes, :);
    plot(nodexy(:,1),nodexy(:,2),'k--')
end
% plot new shape
Magnification = 20; ndivs = 20;
xydisp = [zeros(Nnodes,1) u(1:2:end)] ;
nodesnew = nodes + Magnification*xydisp;
plot(nodesnew(:,1),nodesnew(:,2),'o', ...
    'MarkerEdgeColor','k', 'MarkerFaceColor','r','MarkerSize',10)
hold on; axis equal;
```

Element Calculations

```
function [Kel] = BeamElement2DBE(nodexy, E, A, I)
% This function must return a 4x4 element stiffness matrix: [Kel]
% This matrix must be in the GLOBAL Coordinates
% Input:
% nodexy : [ x1 y1;
%            x2 y2]
% E : Youngs modulus
% I : Second moment of Area

E1 = [ (nodexy(2,1)-nodexy(1,1)) ...
       (nodexy(2,2)-nodexy(1,2)) ];
L = norm(E1);
E1 = E1/L;
E2 = [-E1(2) E1(1)];

Kel = [ ...
    12*E*I/(L^3)   6*E*I/(L^2)  -12*E*I/(L^3)   6*E*I/(L^2)  ; ...
     6*E*I/(L^2)   4*E*I/L       -6*E*I/(L^2)   2*E*I/L      ; ...
   -12*E*I/(L^3)  -6*E*I/(L^2)   12*E*I/(L^3)  -6*E*I/(L^2)  ; ...
     6*E*I/(L^2)   2*E*I/L       -6*E*I/(L^2)   4*E*I/L       ];
```

Assembly and Global solution

Example:



$$K_A = K_B = 1.0e{+}06 * \begin{bmatrix} 1.5000 & 1.5000 & -1.5000 & 1.5000 \\ 1.5000 & 2.0000 & -1.5000 & 1.0000 \\ -1.5000 & -1.5000 & 1.5000 & -1.5000 \\ 1.5000 & 1.0000 & -1.5000 & 2.0000 \end{bmatrix}$$

```
Nodes: (x, y)
0.0    0.0
2.0    0.0
4.0    0.0
Elements: (Node1 Node2), E, A, I,
1 2 2e11  1e-2  5e-6
2 3 2e11  1e-2  5e-6
BCs (Node_number dof specified_disp)
1 1 0
2 1 0
Nodal loads (Node_number dof specified load)
3 1 -5000
```

Assembly of global stiffness matrix:

$$K^G = 1.0e{+}06 * \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{bmatrix} 1.5000 & 1.5000 & -1.5000 & 1.5000 & 0 & 0 \\ 1.5000 & 2.0000 & -1.5000 & 1.0000 & 0 & 0 \\ -1.5000 & -1.5000 & 3.0000 & 0 & -1.5000 & 1.5000 \\ 1.5000 & 1.0000 & 0 & 4.0000 & -1.5000 & 1.0000 \\ 0 & 0 & -1.5000 & -1.5000 & 1.5000 & -1.5000 \\ 0 & 0 & 1.5000 & 1.0000 & -1.5000 & 2.0000 \end{bmatrix}$$

Load:

$$f = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -5000 \\ 0 \end{bmatrix}$$

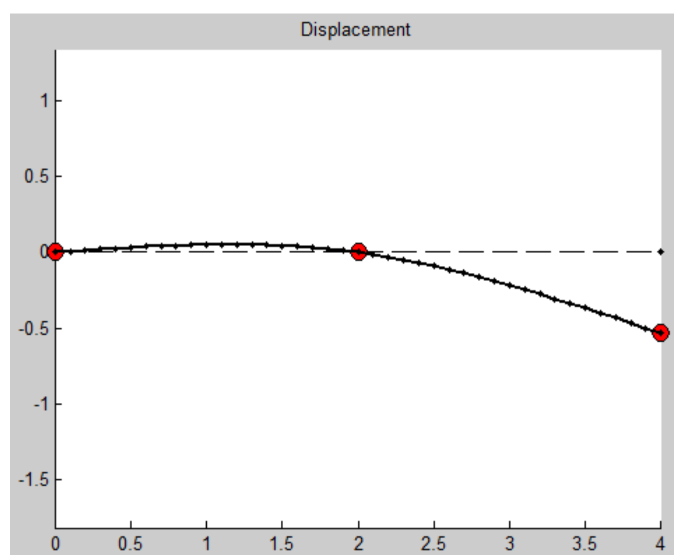
Solution:

Displacement and Rotations :

$$u = \begin{bmatrix} 0 \\ 0.0033 \\ 0 \\ -0.0067 \\ -0.0267 \\ -0.0167 \end{bmatrix}$$

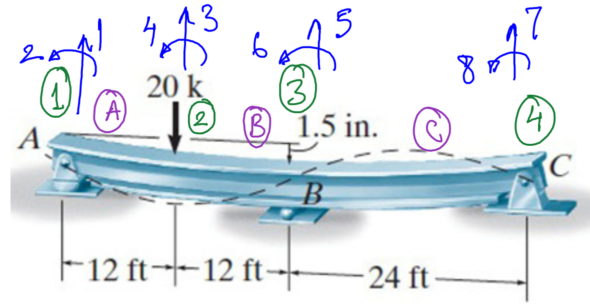
Reactions (Forces and Moments)

$$f = 1.0e{+}04 * \begin{bmatrix} -0.5000 \\ 0 \\ 1.0000 \\ 0 \\ -0.5000 \\ 0 \end{bmatrix}$$



Example

Support B settles by 1.5 in.
Find the reactions and draw the Shear Force and Bending Moment Diagrams of the beam.
E = 29000 ksi ; I = 750 in^4



K1 = K2 =

```
1.0e+03 *
 0.0694    0.4167   -0.0694    0.4167
 0.4167    3.3333   -0.4167    1.6667
-0.0694   -0.4167    0.0694   -0.4167
 0.4167    1.6667   -0.4167    3.3333
```

K3 =

```
1.0e+03 *
 0.0087    0.1042   -0.0087    0.1042
 0.1042    1.6667   -0.1042    0.8333
-0.0087   -0.1042    0.0087   -0.1042
 0.1042    0.8333   -0.1042    1.6667
```

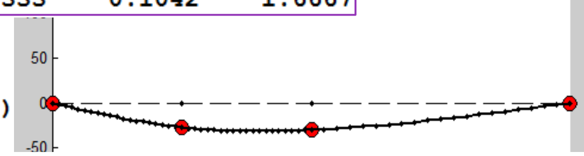
```
Nodes: (x, y)
0.0   0.0
144.0   0.0
288.0   0.0
576.0   0.0
Elements: (Node1 Node2), E, A, I,
1 2 29000 1 750
2 3 29000 1 750
3 4 29000 1 750
BCs (Node_number dof specified_disp)
1 1 0
3 1 -1.5
4 1 0
Nodal loads (Node_number dof specified
2 1 -20
```

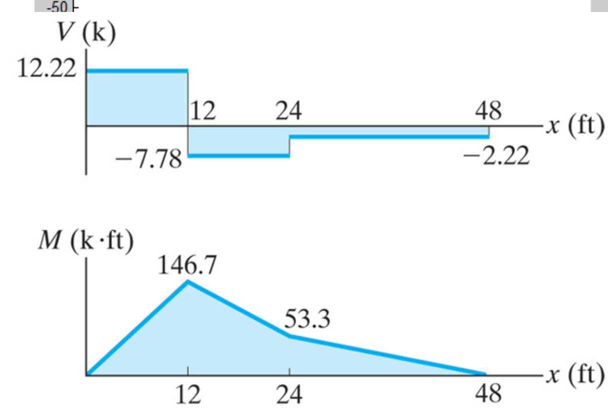
Assembled Kglobal =

```
K =
  1.0e+03 *
   0.0694    0.4167   -0.0694    0.4167         0         0         0         0
   0.4167    3.3333   -0.4167    1.6667         0         0         0         0
  -0.0694   -0.4167    0.1389         0   -0.0694    0.4167         0         0
   0.4167    1.6667         0    6.6667   -0.4167    1.6667         0         0
        0         0   -0.0694   -0.4167    0.0781   -0.3125   -0.0087    0.1042
        0         0    0.4167    1.6667   -0.3125    5.0000   -0.1042    0.8333
        0         0         0         0   -0.0087   -0.1042    0.0087   -0.1042
        0         0         0         0    0.1042    0.8333   -0.1042    1.6667
```

Solution:

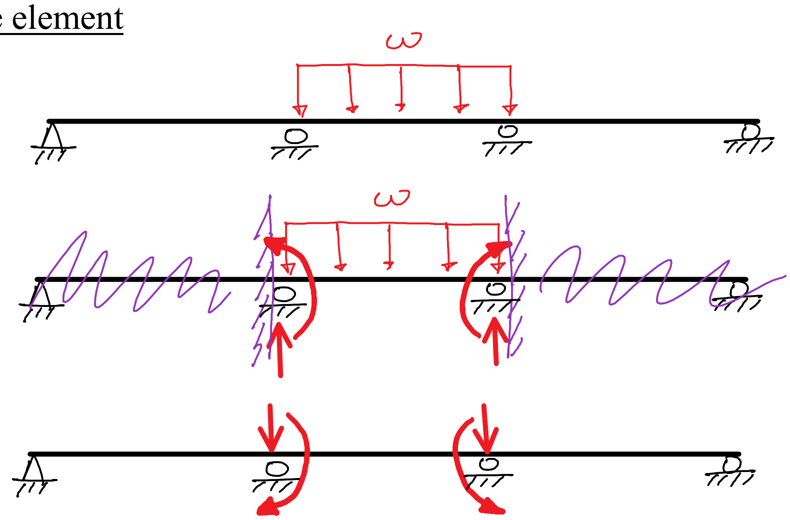
```
Displacement and Rotations :      Reactions (Forces and Moments)
u =                               f =
        0  1                          12.2223
  -0.0114  2                                0
  -1.3602  3                         -20.0000
  -0.0056  4                                0
  -1.5000  5                           5.5555
   0.0024  6                                0
        0  7                           2.2223
   0.0066  8                                0
```





Distributed Loads along the length of the element

Beams with distributed loads along the length can be solved by the stiffness method using fixed-end moments as follows:

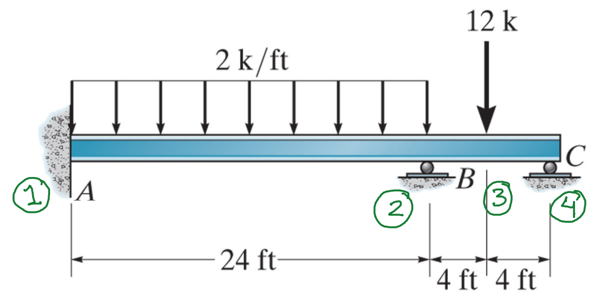


Superimpose with (hand calculation) →

Stiffness Method (MATLAB) →

Example

Determine reactions.
E = 29000 ksi; I = 510in^4



(a)

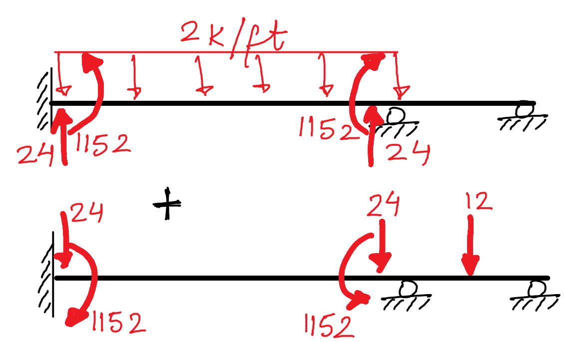
Figure: 15_11aEX04

Dividing the problem into 2 parts :

① Fixed-end Reactions

$$\text{Forces} = \frac{wL}{2} = \frac{2 \times 24}{2} = \underline{24\ k}$$

$$\text{Moments} = \frac{wL^2}{12} = \frac{2 \times 24^2}{12} = 96\ \text{k-ft} = \underline{1152\ \text{k-in}}$$



② Equivalent Nodal loads : (MATLAB)

```
K1 =

   1.0e+05 *

    0.0001    0.0107   -0.0001    0.0107
    0.0107    2.0542   -0.0107    1.0271
   -0.0001   -0.0107    0.0001   -0.0107
    0.0107    1.0271   -0.0107    2.0542
```

K2 = K3 =
```
1.0e+06 *
    0.0016    0.0385   -0.0016    0.0385
    0.0385    1.2325   -0.0385    0.6162
   -0.0016   -0.0385    0.0016   -0.0385
    0.0385    0.6162   -0.0385    1.2325
```

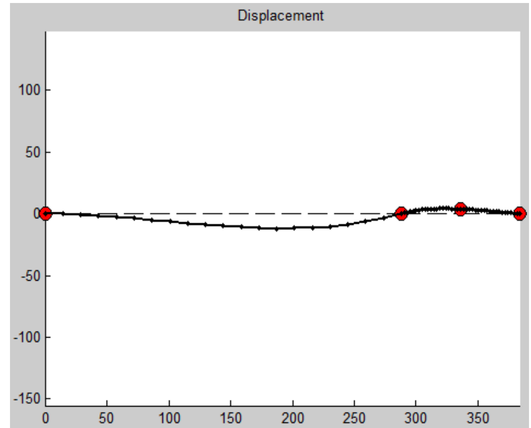
Global system to solve:

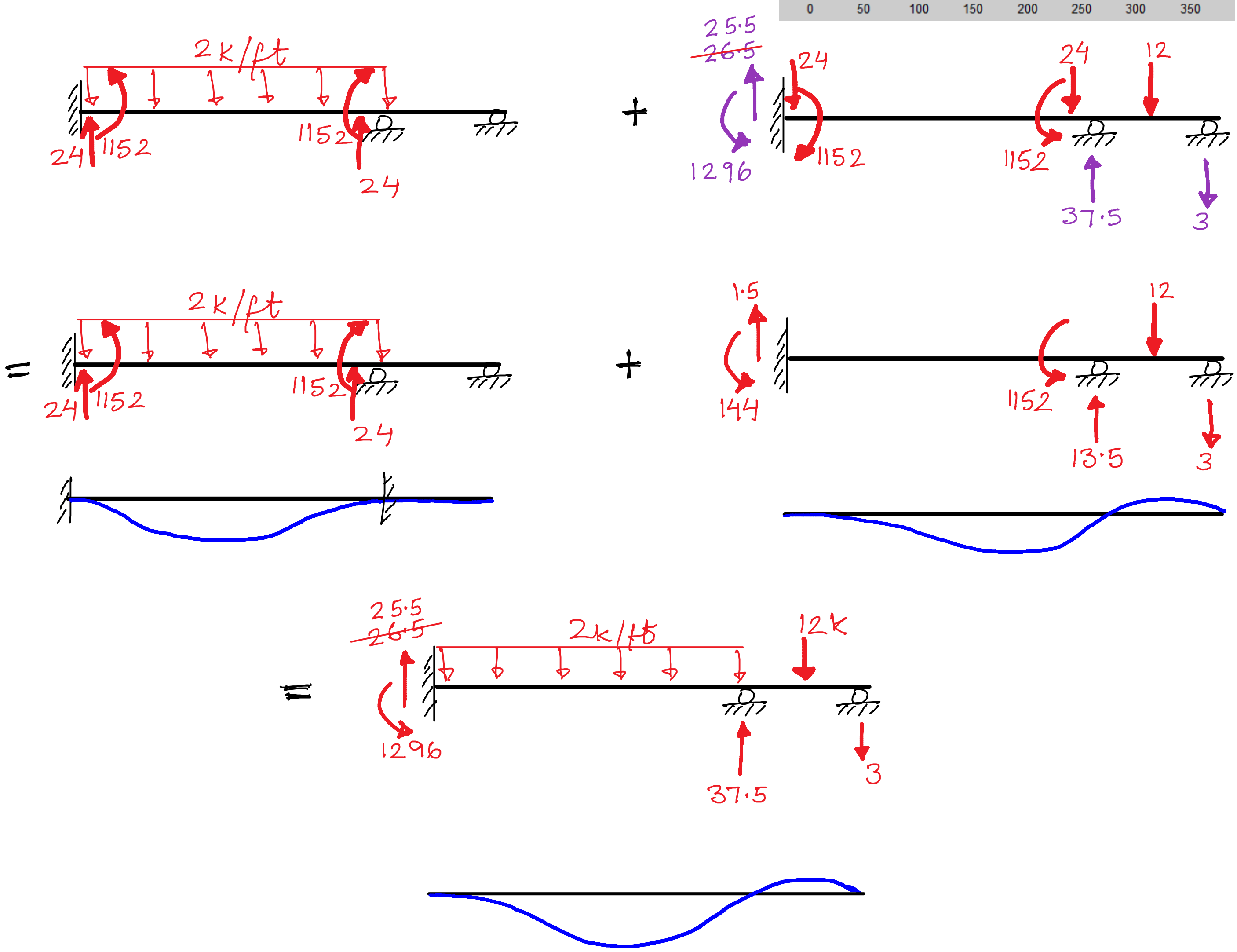
$$1.0e{+}06 * \begin{bmatrix} 0.0000 & 0.0011 & -0.0000 & 0.0011 & 0 & 0 & 0 & 0 \\ 0.0011 & 0.2054 & -0.0011 & 0.1027 & 0 & 0 & 0 & 0 \\ -0.0000 & -0.0011 & 0.0016 & 0.0374 & -0.0016 & 0.0385 & 0 & 0 \\ 0.0011 & 0.1027 & 0.0374 & 1.4379 & -0.0385 & 0.6162 & 0 & 0 \\ 0 & 0 & -0.0016 & -0.0385 & 0.0032 & 0 & -0.0016 & 0.0385 \\ 0 & 0 & 0.0385 & 0.6162 & 0 & 2.4650 & -0.0385 & 0.6162 \\ 0 & 0 & 0 & 0 & -0.0016 & -0.0385 & 0.0016 & -0.0385 \\ 0 & 0 & 0 & 0 & 0.0385 & 0.6162 & -0.0385 & 1.2325 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \theta 2 \\ \Delta 3 \\ \theta 3 \\ 0 \\ \theta 4 \end{bmatrix} = \begin{bmatrix} -24 \\ -1152 \\ -24 \\ 1152 \\ -12 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} Y1 \\ M1 \\ Y2 \\ 0 \\ 0 \\ 0 \\ Y4 \\ 0 \end{bmatrix}$$

Solution to Part ②

```
Displacement and Rotations :
u =
         0
         0
         0
    0.0014
    0.0187
   -0.0002
         0
   -0.0005
```

```
Reactions (Forces and Moments)
f =
   1.0e+03 *
    0.0015
    0.1440
    0.0135
    1.1520
   -0.0120
         0
   -0.0030
         0
```

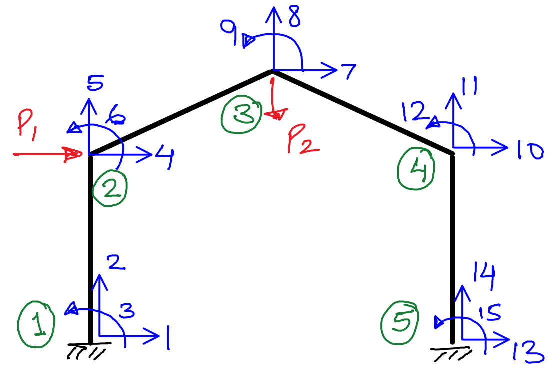




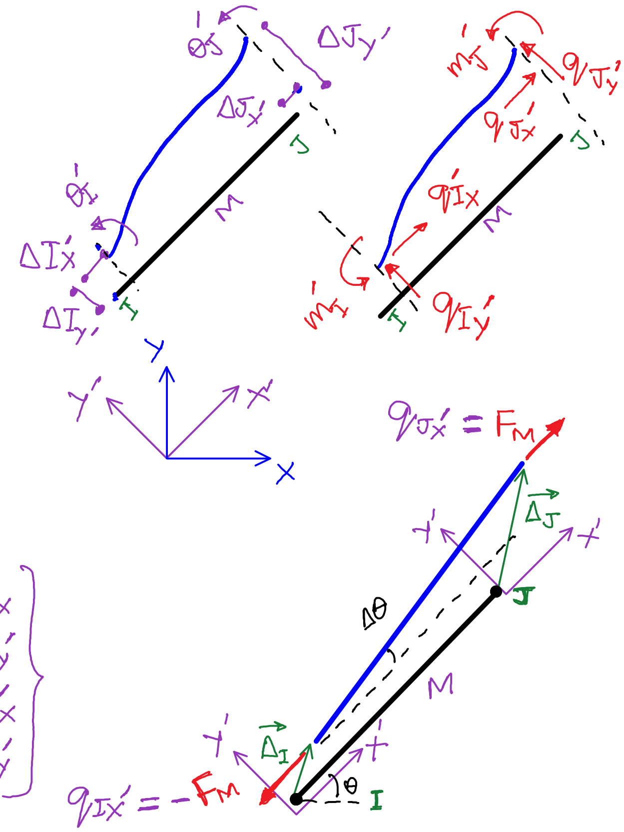
<u>Stiffness Method for Frame Structures</u>

For frame problems (with possibly inclined beam elements), the stiffness method can be used to solve the problem by <u>transforming element stiffness matrices</u> from the LOCAL to GLOBAL coordinates.

Note that in addition to the usual bending terms, we will also have to account for <u>axial effects</u>. These axial effects can be accounted for by simply treating the beam element as a truss element in the axial direction.



Consider a frame element in <u>local</u> $(x'y')$ coordinates:



The force-displacement relationships in <u>local</u> co-ordinates can be written by combining beam & truss elements:

$$\begin{Bmatrix} q'_{Ix} \\ 0 \\ q'_{Jx} \\ 0 \end{Bmatrix} = \begin{bmatrix} \frac{AE}{L} & 0 & -\frac{AE}{L} & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{AE}{L} & 0 & \frac{AE}{L} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \Delta'_{Ix} \\ \Delta'_{Iy} \\ \Delta'_{Jx} \\ \Delta'_{Jy} \end{Bmatrix}$$

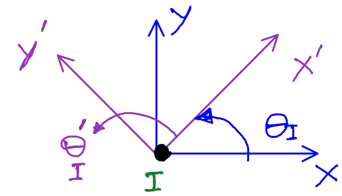
$$\begin{Bmatrix} q_{Ix'} \\ q'_{Iy} \\ m'_I \\ q'_{Jx} \\ q'_{Jy} \\ m'_J \end{Bmatrix} = \left[\begin{array}{c|cc|c|cc} AE/L & 0 & 0 & -AE/L & 0 & 0 \\ \hline 0 & 12EI/L^3 & \frac{6EI}{L^2} & 0 & -12EI/L^3 & \frac{6EI}{L^2} \\ 0 & 6EI/L^2 & \frac{4EI}{L} & 0 & -6EI/L^2 & \frac{2EI}{L} \\ \hline -AE/L & 0 & 0 & AE/L & 0 & 0 \\ \hline 0 & -12EI/L^3 & -\frac{6EI}{L^2} & 0 & 12EI/L^3 & -\frac{6EI}{L^2} \\ 0 & 6EI/L^2 & \frac{2EI}{L} & 0 & -6EI/L^2 & \frac{4EI}{L} \end{array}\right] \begin{Bmatrix} \Delta_{Ix'} \\ \Delta I'_y \\ \theta'_I \\ \Delta J_{x'} \\ \Delta J_{y'} \\ \theta'_J \end{Bmatrix}$$

$$\boxed{q'_M = \underset{\sim}{K}'_M \, d'_M} \qquad (2)$$

Transformation from Local to Global coordinates

Each node has 3 degrees of freedom: (X, Y, θ_I) or (X', Y', θ_I')

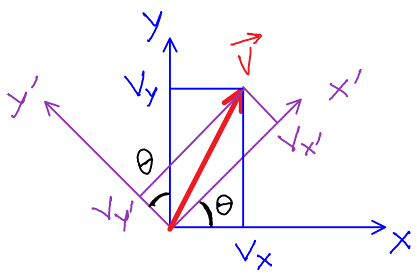
But $\theta_I' = \theta_I$



Note:

$$V_x = V_{x'}\cos(\theta) - V_{y'}\sin(\theta)$$
$$V_y = V_{x'}\sin(\theta) + V_{y'}\cos(\theta)$$
$$\theta_I = \theta_I'$$

Thus transformation rules derived earlier for truss members between (X, Y) and (X', Y') still hold:



$$\vec{V} = \{V_x, V_y\}_{x,y}$$
$$= \{V_{x'}, V_{y'}\}_{x',y'}$$

In matrix form:

$$\begin{Bmatrix} V_x \\ V_y \\ \theta \end{Bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\underset{\sim}{T} \,=\, \mathtt{Qrot}^T} \begin{Bmatrix} V_{x'} \\ V_{y'} \\ \theta' \end{Bmatrix}$$

Reverse:

$$\begin{Bmatrix} V_{x'} \\ V_{y'} \\ \theta_I' \end{Bmatrix} = \underbrace{\begin{bmatrix} \underset{\sim}{T}^T \end{bmatrix}}_{=\,\mathtt{Qrot}} \begin{Bmatrix} V_x \\ V_y \\ \theta_I \end{Bmatrix}$$

Note:

Transformation matrix ***T*** defined above is the same as $\mathtt{Qrot}^T$ defined in the provided MATLAB code.

Converting Local co-ordinates to Global:

$$\begin{Bmatrix} q_{Ix'} \\ q_{Iy'} \\ m_I' \\ \hline q_{Jx}' \\ q_{Jy}' \\ m_J' \end{Bmatrix} = \left[\begin{array}{c|c} \underset{\sim}{T}_M^T \ (\mathtt{Qrot}) & \\ \hline & \underset{\sim}{T}_M^T \ (\mathtt{Qrot}) \end{array}\right] \begin{Bmatrix} q_{Ix} \\ q_{Iy} \\ m_I \\ \hline q_{Jx} \\ q_{Jy} \\ m_J \end{Bmatrix} \quad \text{—— (1)}$$

$$\begin{Bmatrix} \Delta_{Ix'} \\ \Delta_{Iy'} \\ \theta_I' \\ \hline \Delta_{Jx'} \\ \Delta_{Jy'} \\ \theta_J' \end{Bmatrix} = \left[\begin{array}{c|c} \underset{\sim}{T}_M^T \ (\mathtt{Qrot}) & \\ \hline & \underset{\sim}{T}_M^T \ (\mathtt{Qrot}) \end{array}\right] \begin{Bmatrix} \Delta_{Ix} \\ \Delta_{Iy} \\ \theta_I \\ \hline \Delta_{Jx} \\ \Delta_{Jy} \\ \theta_J \end{Bmatrix} \quad \text{—— (2)}$$

Element Stiffness Matrix in GLOBAL coordinates:

Substituting the transformation relations (l) and (2) into LOCAL force (moment) - displacement (rotation) relationships (L):

$$\underset{\sim}{q}'_M = \underset{\sim}{K}'_M \, \underset{\sim}{d}'_M$$

$$\begin{bmatrix} \underset{\sim}{T}_M^T & 0 \\ 0 & \underset{\sim}{T}_M^T \end{bmatrix} \underline{q}_M = \underset{\sim}{K}'_M \begin{bmatrix} \underset{\sim}{T}_M^T & 0 \\ 0 & \underset{\sim}{T}_M^T \end{bmatrix} \underline{d}_M$$

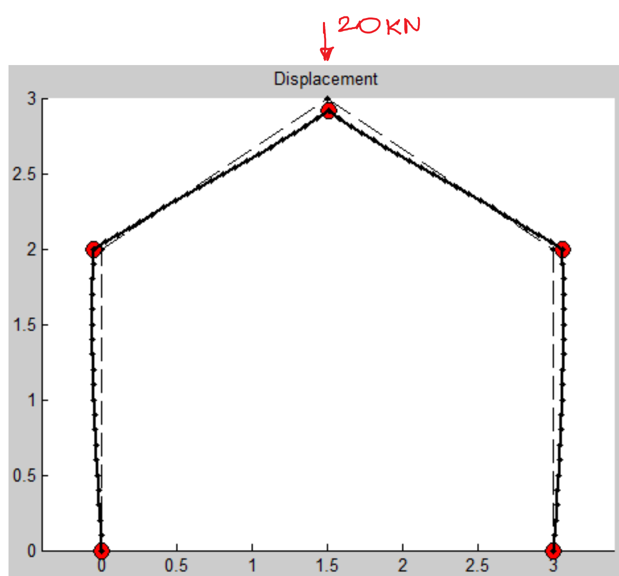
$$\begin{Bmatrix} q_{Ix} \\ q_{Iy} \\ m_I \\ \hline q_{Jx} \\ q_{Jy} \\ m_J \end{Bmatrix} = \underbrace{\overbrace{\left[\begin{array}{c|c} \underset{\sim}{T}_M \ (\texttt{Qrot}^{\texttt{T}}) & \\ \hline & \underset{\sim}{T}_M \ (\texttt{Qrot}^{\texttt{T}}) \end{array}\right]}^{\texttt{Tmatrix}^{\texttt{T}} \text{ (in MATLAB code)}} \left[\underset{\sim}{K}'_M \right] \overbrace{\left[\begin{array}{c|c} \underset{\sim}{T}_M^T \ (\texttt{Qrot}) & \\ \hline & \underset{\sim}{T}_M^T \ (\texttt{Qrot}) \end{array}\right]}^{\texttt{Tmatrix} \text{ (in MATLAB code)}}}_{\left[\underset{\sim}{K}_M\right]} \begin{Bmatrix} \Delta_{Ix} \\ \Delta_{Iy} \\ \theta_I \\ \hline \Delta_{Jx} \\ \Delta_{Jy} \\ \theta_J \end{Bmatrix}$$

Thus, similar to trusses:

$$\underset{(\text{global})}{\left[\underset{\sim}{K}_M\right]} = \begin{bmatrix} \underset{\sim}{T}_M & 0 \\ 0 & \underset{\sim}{T}_M \end{bmatrix} \underset{(\text{local})}{\left[\underset{\sim}{K}'_M\right]} \begin{bmatrix} \underset{\sim}{T}_M^T & 0 \\ 0 & \underset{\sim}{T}'_M \end{bmatrix}$$

Example:

```
Nodes: (x, y)
0.0   0.0
0.0   2.0
1.5   3.0
3.0   2.0
3.0   0.0
Elements: (Node1 Node2), E, A, I,
1 2 2e11  1e-2  5e-6
2 3 2e11  1e-2  5e-6
3 4 2e11  1e-2  5e-6
4 5 2e11  1e-2  5e-6
BCs (Node_number dof specified_disp)
1 1 0
1 2 0
5 1 0
5 2 0
Nodal loads (Node_number dof specified load)
3 2 -20000
```



Frame 2D MATLAB Code:

```matlab
% Main code for solving 2D Frame problems using Stiffness method

clear all; clc; close all; % clear all the existing variables (new start)

% Obtain the input file name from the user & Read Input
inpfilename = uigetfile('*.txt','Select the input file');

[nodes, elems, E, A, I, bcs, loads] = getframedata2D(inpfilename);
Nel = size(elems,1);
Nnodes = size(nodes,1);

% Decide degrees of freedom + Initialize Matrices
alldofs = 1:3*Nnodes;
K = zeros(3*Nnodes);
u = zeros(3*Nnodes,1);
f = zeros(3*Nnodes,1);
% Note: Degrees of Freedom correspoding to node "i"
% are [3*(i-1)+1   3*(i-1)+2  3*(i-1)+3]

% Boundary conditions
dofspec = [];
for ii = 1:size(bcs,1)
    thisdof = 3*(bcs(ii,1)-1)+bcs(ii,2);
    dofspec = [dofspec thisdof];
    u(thisdof) = bcs(ii,3);
end
doffree = alldofs;
doffree(dofspec) = []; % Delete specified dofs from All dofs

% Nodal Loads
for ii = 1:size(loads,1)
    f(3*(loads(ii,1)-1)+loads(ii,2)) = loads(ii,3);
end

% Initialize the global stiffness matrix
for iel = 1:Nel
    elnodes = elems( iel, 1:2);
    nodexy = nodes(elnodes, :);

    % Get the element stiffness matrix for the current element
    [Kel] = FrameElement2DBE(nodexy, E(iel), A(iel), I(iel));

    % Assemble the element stiffness matrix into the global stiffness matrix K
    eldofs = 3*(elnodes(1)-1)+1:3*elnodes(1);
    eldofs = [eldofs 3*(elnodes(2)-1)+1:3*elnodes(2)];
    K(eldofs,eldofs) = K(eldofs,eldofs) + Kel;
end

% Solve
u(doffree) = K(doffree,doffree)\(f(doffree)-K(doffree,dofspec)*u(dofspec));
f(dofspec) = K(dofspec,:)*u;
% format long
disp(['Displacement and Rotations :']); u
disp(['Reactions (Forces and Moments)']); f
```

Plotting

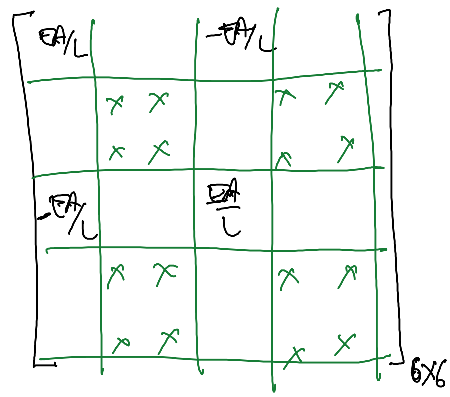
```
% plot old shape
figure(1); hold on;
plot(nodes(:,1),nodes(:,2),'k.')
hold on; axis equal;
for iel = 1:Nel
    elnodes = elems(iel, 1:2);
    nodexy = nodes(elnodes, :);
    plot(nodexy(:,1),nodexy(:,2),'k--')
end
% plot new shape
Magnification = 20; ndivs = 20;
xydisp = [u(1:3:end) u(2:3:end)] ;
nodesnew = nodes + Magnification*xydisp;
plot(nodesnew(:,1),nodesnew(:,2),'o', ...
    'MarkerEdgeColor','k', 'MarkerFaceColor','r','MarkerSize',10)
hold on; axis equal;

for iel = 1:Nel
    elnodes = elems(iel, 1:2);
    E1 = [ (nodes(elnodes(2),1)-nodes(elnodes(1),1)) ...
           (nodes(elnodes(2),2)-nodes(elnodes(1),2)) ];
    le = norm(E1);
    E1 = E1/le;
    E2 = [-E1(2) E1(1)];

    eldofs = 3*(elnodes(1)-1)+1:3*elnodes(1);
    eldofs = [eldofs 3*(elnodes(2)-1)+1:3*elnodes(2)];
    eldisp = u(eldofs);

    Qrot = [E1; E2]; % Transforms global to element d_E = Q d_G
    Qrot(3,3) = 1;
    Tmatrix = [Qrot zeros(3); zeros(3) Qrot];
    eldispLOC = Tmatrix*eldisp;
    for jj = 1:ndivs+1
        xi = (jj-1)/ndivs;
        xdispLOC = eldispLOC(1)*(1-xi)+eldispLOC(4)*xi;
        ydispLOC = eldispLOC(2)*(1-3*xi^2+2*xi^3)+eldispLOC(5)*(3*xi^2-2*xi^3) ...
            + eldispLOC(3)*le*(xi-2*xi^2+xi^3) + eldispLOC(6)*le*(-xi^2+xi^3);

        xydisp = (Qrot([1,2],[1,2]))'*[xdispLOC ; ydispLOC];
        plotpts(jj,1) = nodes(elnodes(1),1) + xi*le*E1(1) + Magnification*xydisp(1);
        plotpts(jj,2) = nodes(elnodes(1),2) + xi*le*E1(2) + Magnification*xydisp(2);
    end
    plot(plotpts(:,1),plotpts(:,2),'k.-','LineWidth',2)
end
title('Displacement');
```

Frame Element Code

```
function [Kel] = FrameElement2DBE(nodexy, E, A, I)
% This function must return a 4x4 element stiffness matrix: [Kel]
% This matrix must be in the GLOBAL Coordinates
% Input:
% nodexy : [ x1 y1;
%            x2 y2]
% E : Youngs modulus
% I : Second moment of Area

E1 = [ (nodexy(2,1)-nodexy(1,1)) ...
       (nodexy(2,2)-nodexy(1,2)) ];
L = norm(E1);
E1 = E1/L;
E2 = [-E1(2) E1(1)];

Kel_bend = [ ...
   12*E*I/(L^3)   6*E*I/(L^2)   -12*E*I/(L^3)   6*E*I/(L^2)  ; ...
   6*E*I/(L^2)    4*E*I/L       -6*E*I/(L^2)    2*E*I/L      ; ...
  -12*E*I/(L^3)  -6*E*I/(L^2)    12*E*I/(L^3)  -6*E*I/(L^2)  ; ...
   6*E*I/(L^2)    2*E*I/L       -6*E*I/(L^2)    4*E*I/L       ];
Kel_axial = E*A/L*[1 -1; -1 1];

Kel_LOC([1,4],[1,4]) = Kel_axial;
Kel_LOC([2,3,5,6],[2,3,5,6]) = Kel_bend;

Qrot = [E1; E2]; % Transforms global to element d_E = Q d_G
Qrot(3,3) = 1;
Tmatrix = [Qrot zeros(3); zeros(3) Qrot];

Kel = Tmatrix'*Kel_LOC*Tmatrix;
```

$$\vec{E1} = \vec{n} = \left[(x_J - x_I)\hat{i}' + (y_J - y_I)\hat{j}'\right]/L$$

$$L = \sqrt{(x_J - x_I)^2 + (y_J - y_I)^2}$$

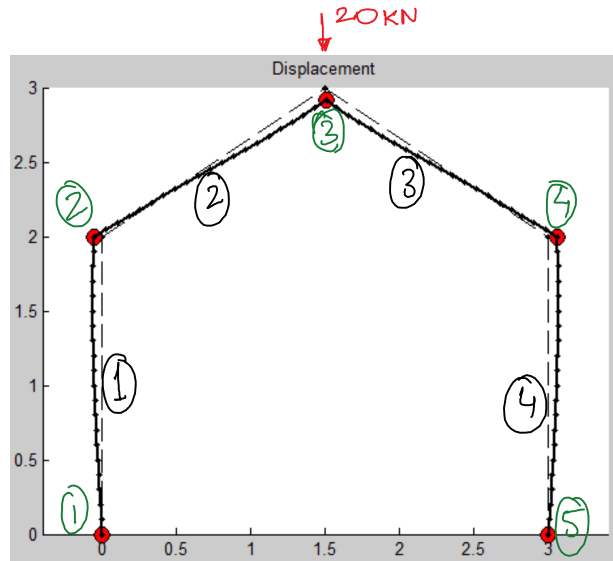


$$Q_{rot} = \begin{bmatrix} \vec{E1} \\ \vec{E2} \end{bmatrix} = \begin{bmatrix} \overset{C}{(x_J - x_I)/L} & \overset{S}{(y_J - y_I)/L} \\ \underset{-S}{-(y_J - y_I)/L} & \underset{C}{(x_J - x_I)/L} \end{bmatrix}$$

$$\text{Tmatrix} = \left[\begin{array}{c|c} \underset{\sim}{Q}_{3\times 3} & \underset{\sim}{0}_{3\times 3} \\ \hline 0_{3\times 3} & \underset{\sim}{Q}_{3\times 3} \end{array}\right]$$

Example:

```
Nodes: (x, y)
0.0   0.0
0.0   2.0
1.5   3.0
3.0   2.0
3.0   0.0
Elements: (Node1 Node2), E, A, I,
1 2 2e11  1e-2  5e-6
2 3 2e11  1e-2  5e-6
3 4 2e11  1e-2  5e-6
4 5 2e11  1e-2  5e-6
BCs (Node_number dof specified_disp)
1 1 0
1 2 0
5 1 0
5 2 0
Nodal loads (Node_number dof specified load)
3 2 -20000
```



Element Stiffness matrices:

(local coordinates)

K1L =
1.0e+09 *

1.0000	0	0	-1.0000	0	0
0	0.0015	0.0015	0	-0.0015	0.0015
0	0.0015	0.0020	0	-0.0015	0.0010
-1.0000	0	0	1.0000	0	0
0	-0.0015	-0.0015	0	0.0015	-0.0015
0	0.0015	0.0010	0	-0.0015	0.0020

K2L =
1.0e+09 *

1.1094	0	0	-1.1094	0	0
0	0.0020	0.0018	0	-0.0020	0.0018
0	0.0018	0.0022	0	-0.0018	0.0011
-1.1094	0	0	1.1094	0	0
0	-0.0020	-0.0018	0	0.0020	-0.0018
0	0.0018	0.0011	0	-0.0018	0.0022

K3L =
1.0e+09 *

1.1094	0	0	-1.1094	0	0
0	0.0020	0.0018	0	-0.0020	0.0018
0	0.0018	0.0022	0	-0.0018	0.0011
-1.1094	0	0	1.1094	0	0
0	-0.0020	-0.0018	0	0.0020	-0.0018
0	0.0018	0.0011	0	-0.0018	0.0022

K4L =
1.0e+09 *

1.0000	0	0	-1.0000	0	0
0	0.0015	0.0015	0	-0.0015	0.0015
0	0.0015	0.0020	0	-0.0015	0.0010
-1.0000	0	0	1.0000	0	0
0	-0.0015	-0.0015	0	0.0015	-0.0015
0	0.0015	0.0010	0	-0.0015	0.0020

(Global Co-ordinates)

K1 =
1.0e+09 *

0.0015	0	-0.0015	-0.0015	0	-0.0015
0	1.0000	0	0	-1.0000	0
-0.0015	0	0.0020	0.0015	0	0.0010
-0.0015	0	0.0015	0.0015	0	0.0015
0	-1.0000	0	0	1.0000	0
-0.0015	0	0.0010	0.0015	0	0.0020

K2 =
1.0e+08 *

7.6868	5.1109	-0.0102	-7.6868	-5.1109	-0.0102
5.1109	3.4277	0.0154	-5.1109	-3.4277	0.0154
-0.0102	0.0154	0.0222	0.0102	-0.0154	0.0111
-7.6868	-5.1109	0.0102	7.6868	5.1109	0.0102
-5.1109	-3.4277	-0.0154	5.1109	3.4277	-0.0154
-0.0102	0.0154	0.0111	0.0102	-0.0154	0.0222

K3 =
1.0e+08 *

7.6868	-5.1109	0.0102	-7.6868	5.1109	0.0102
-5.1109	3.4277	0.0154	5.1109	-3.4277	0.0154
0.0102	0.0154	0.0222	-0.0102	-0.0154	0.0111
-7.6868	5.1109	-0.0102	7.6868	-5.1109	-0.0102
5.1109	-3.4277	-0.0154	-5.1109	3.4277	-0.0154
0.0102	0.0154	0.0111	-0.0102	-0.0154	0.0222

K4 =
1.0e+09 *

0.0015	0	0.0015	-0.0015	0	0.0015
0	1.0000	0	0	-1.0000	0
0.0015	0	0.0020	-0.0015	0	0.0010
-0.0015	0	-0.0015	0.0015	0	-0.0015
0	-1.0000	0	0	1.0000	0
0.0015	0	0.0010	-0.0015	0	0.0020

Global structural stiffness matrix (15 × 15) :

K =

1.0e+09 *

Columns 1 through 8 — Columns 9 through 15

$$
\begin{bmatrix}
0.0015 & 0 & -0.0015 & -0.0015 & 0 & -0.0015 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.0000 & 0 & 0 & -1.0000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.0015 & 0 & 0.0020 & 0.0015 & 0 & 0.0010 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.0015 & 0 & 0.0015 & 0.7702 & 0.5111 & 0.0005 & -0.7687 & -0.5111 & -0.0010 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1.0000 & 0 & 0.5111 & 1.3428 & 0.0015 & -0.5111 & -0.3428 & 0.0015 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.0015 & 0 & 0.0010 & 0.0005 & 0.0015 & 0.0042 & 0.0010 & -0.0015 & 0.0011 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -0.7687 & -0.5111 & 0.0010 & 1.5374 & 0 & 0.0020 & -0.7687 & 0.5111 & 0.0010 & 0 & 0 & 0 \\
0 & 0 & 0 & -0.5111 & -0.3428 & -0.0015 & 0 & 0.6855 & 0 & 0.5111 & -0.3428 & 0.0015 & 0 & 0 & 0 \\
0 & 0 & 0 & -0.0010 & 0.0015 & 0.0011 & 0.0020 & 0 & 0.0044 & -0.0010 & -0.0015 & 0.0011 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -0.7687 & 0.5111 & -0.0010 & 0.7702 & -0.5111 & 0.0005 & -0.0015 & 0 & 0.0015 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.5111 & -0.3428 & -0.0015 & -0.5111 & 1.3428 & -0.0015 & 0 & -1.0000 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.0010 & 0.0015 & 0.0011 & 0.0005 & -0.0015 & 0.0042 & -0.0015 & 0 & 0.0010 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.0015 & 0 & -0.0015 & 0.0015 & 0 & -0.0015 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.0000 & 0 & 0 & 1.0000 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0015 & 0 & 0.0010 & -0.0015 & 0 & 0.0020
\end{bmatrix}
$$

Solution:

Displacement and Rotations :

u =

$$
\begin{array}{cl}
\begin{bmatrix} 0 \\ 0 \\ 0.0031 \\ -0.0029 \\ -0.0000 \\ -0.0020 \\ 0.0000 \\ -0.0043 \\ 0.0000 \\ 0.0029 \\ -0.0000 \\ 0.0020 \\ 0 \\ 0 \\ -0.0031 \end{bmatrix} &
\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \end{matrix}
\end{array}
$$

Reactions (Forces and Moments)

f =

1.0e+04 *

$$
\begin{array}{cl}
\begin{bmatrix} 0.2560 \\ 1.0000 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -2.0000 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0.2560 \\ 1.0000 \\ 0 \end{bmatrix} &
\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \end{matrix}
\end{array}
$$