CEE6501 — Lecture 7.2

Global Formulation for 2D Frame Analysis

Learning Objectives

By the end of this lecture, you will be able to:

- Define consistent local and global coordinate systems for 2D frame elements
- Build the 2D frame transformation matrix $\boldsymbol{T}$ and apply it to forces and displacements
- Derive and use the global member relation $\boldsymbol{F} = \boldsymbol{K}\boldsymbol{u} + \boldsymbol{F}^F$
- Compute and transform fixed-end force (FEF) vectors for member loads
- Outline and execute the DSM forward pass (solve) and backward pass (recover element forces)

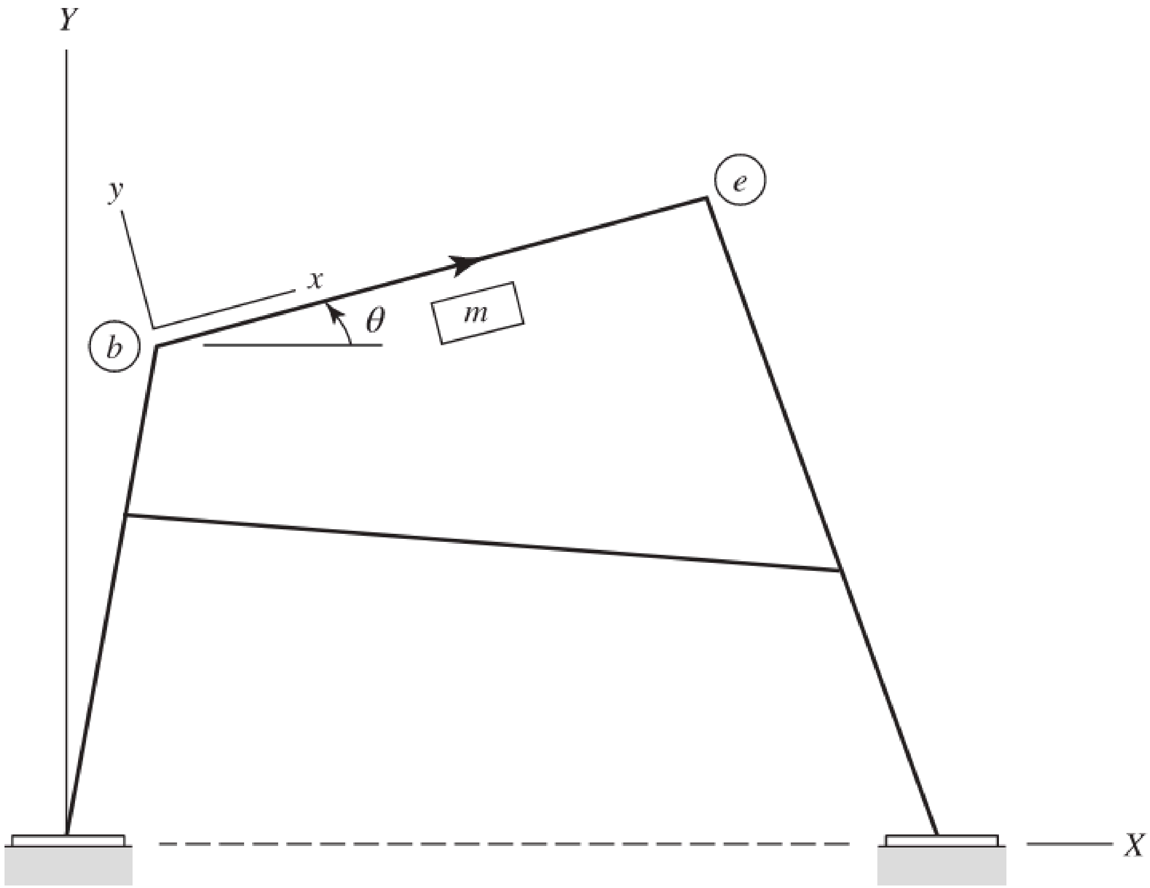
Agenda

Part 1 — Local ↔ global transformations
Part 2 — Global stiffness relationship
Part 3 — 2D frame element stiffness matrix (global form)
Part 4 — Member forces (global coordinates)
Part 5 — Worked example (Kassimali 6.3–6.4)
Part 6 — DSM full procedure for 2D frames

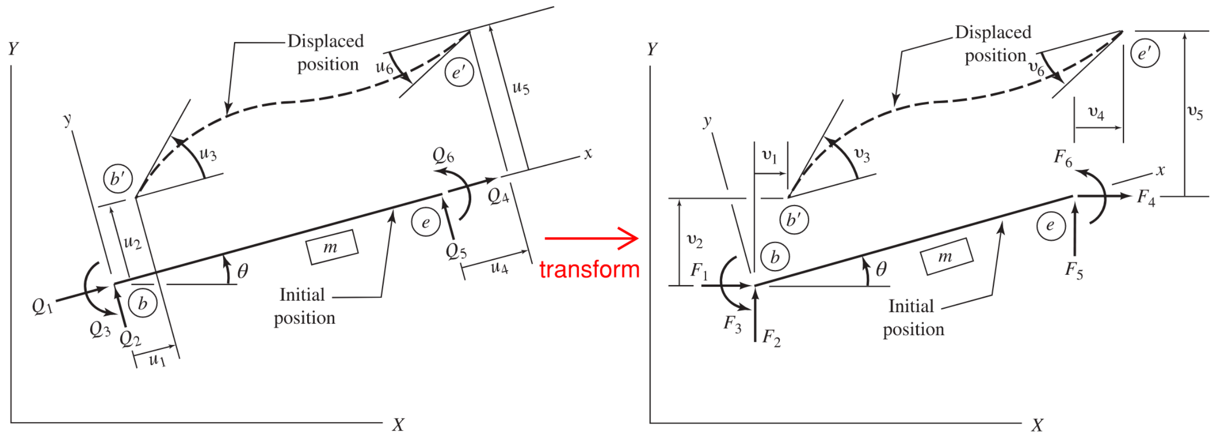
Part 1 — Local to Global Transformation

Note: See Lecture 3.2 (Trusses) for detailed background on direction cosines and transformations.

Generic Element in a Structure



Local vs. Global Perspective



- **Local coordinates (left):** forces $\boldsymbol{Q}$, displacements $\boldsymbol{u}$
- **Global coordinates (right):** forces $\boldsymbol{F}$, displacements $\boldsymbol{v}$

Global ($\boldsymbol{F}$) → Local ($\boldsymbol{Q}$) Force Transformation

Use trigonometric relationships to resolve global forces into the local coordinate system.

At node i:

$$\begin{aligned} Q_1 &= F_1 \cos\theta + F_2 \sin\theta \\ Q_2 &= -F_1 \sin\theta + F_2 \cos\theta \\ Q_3 &= F_3 \end{aligned}$$

The local and global z-axes are aligned (out of plane), so the bending moment is unchanged:

$$Q_3 = F_3$$

At node j:

$$\begin{aligned} Q_4 &= F_4 \cos\theta + F_5 \sin\theta \\ Q_5 &= -F_4 \sin\theta + F_5 \cos\theta \\ Q_6 &= F_6 \end{aligned}$$

Global ($\boldsymbol{F}$) → Local ($\boldsymbol{Q}$) Force Transformation

- Local element forces $\boldsymbol{Q}$ are obtained by **rotating** global nodal forces $\boldsymbol{F}$ into the member's local coordinate system
- Each node contributes a 3×3 transformation block:
  - Translational DOFs → **rotated by** θ
  - Rotational DOF → **invariant** (local and global z-axes are aligned)

$$\boldsymbol{Q} = \boldsymbol{T}\,\boldsymbol{F}$$

$$\begin{Bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \\ Q_6 \end{Bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\theta & \sin\theta & 0 \\ 0 & 0 & 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{T}} \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{Bmatrix}$$

Note: This matches the truss transformation for translational DOFs; the only addition is the rotational DOF (unchanged).

Global ($\boldsymbol{v}$) → Local ($\boldsymbol{u}$) Displacements

- Displacements are transformed using the **same rotation matrix** as forces
- Forces and displacements transform identically because they are defined along the **same directions**

$$\boldsymbol{u} = \boldsymbol{T}\boldsymbol{v}$$

Local ($\boldsymbol{Q}$) → Global ($\boldsymbol{F}$)

- Reverse of the global → local process
- Rotate local quantities back into the global (X, Y) directions

$$\boldsymbol{F} = \boldsymbol{T}^{\top}\boldsymbol{Q}$$

Local ($\boldsymbol{u}$) → Global ($\boldsymbol{v}$)

- Reverse of the global → local process
- Rotate local quantities back into the global (X, Y) directions

$$\boldsymbol{v} = \boldsymbol{T}^{\top}\boldsymbol{u}$$

Part 2 — Global Stiffness Relationship

We start from the local relation:

$$\mathbf{Q} = \mathbf{k}\mathbf{u} + \mathbf{Q}^F$$

where:

- $\mathbf{Q}$ = local end force vector
- $\mathbf{k}$ = local stiffness matrix
- $\mathbf{u}$ = local displacement vector
- $\mathbf{Q}^F$ = local fixed-end force vector

Step 1 — Transform Forces to the Global System

Forces transform as:

$$\mathbf{F} = \mathbf{T}^\mathsf{T}\mathbf{Q}$$

Substitute the local stiffness relation:

$$\mathbf{F} = \mathbf{T}^\mathsf{T}(\mathbf{k}\mathbf{u} + \mathbf{Q}^F)$$

Distribute:

$$\mathbf{F} = \mathbf{T}^\mathsf{T}\mathbf{k}\mathbf{u} + \mathbf{T}^\mathsf{T}\mathbf{Q}^F$$

Step 2 — Transform Displacements

Displacements transform as:

$$\mathbf{u} = \mathbf{T}\mathbf{v}$$

Substitute into the previous equation:

$$\mathbf{F} = \mathbf{T}^{\mathsf{T}}\mathbf{k}\mathbf{T}\mathbf{v} + \mathbf{T}^{\mathsf{T}}\mathbf{Q}^{F}$$

Step 3 — Define Global Quantities

Global member stiffness matrix:

$$\boxed{\mathbf{K} = \mathbf{T}^{\mathsf{T}}\mathbf{k}\mathbf{T}}$$

Global fixed-end force vector:

$$\boxed{\mathbf{F}^{F} = \mathbf{T}^{\mathsf{T}}\mathbf{Q}^{F}}$$

Final Global Member Relation

Substitute the definitions:

$$\boxed{\mathbf{F} = \mathbf{K}\mathbf{v} + \mathbf{F}^F}$$

In our course notation, we typically use $\mathbf{u}$ for global displacements:

$$\boxed{\mathbf{F} = \mathbf{K}\mathbf{u} + \mathbf{F}^F}$$

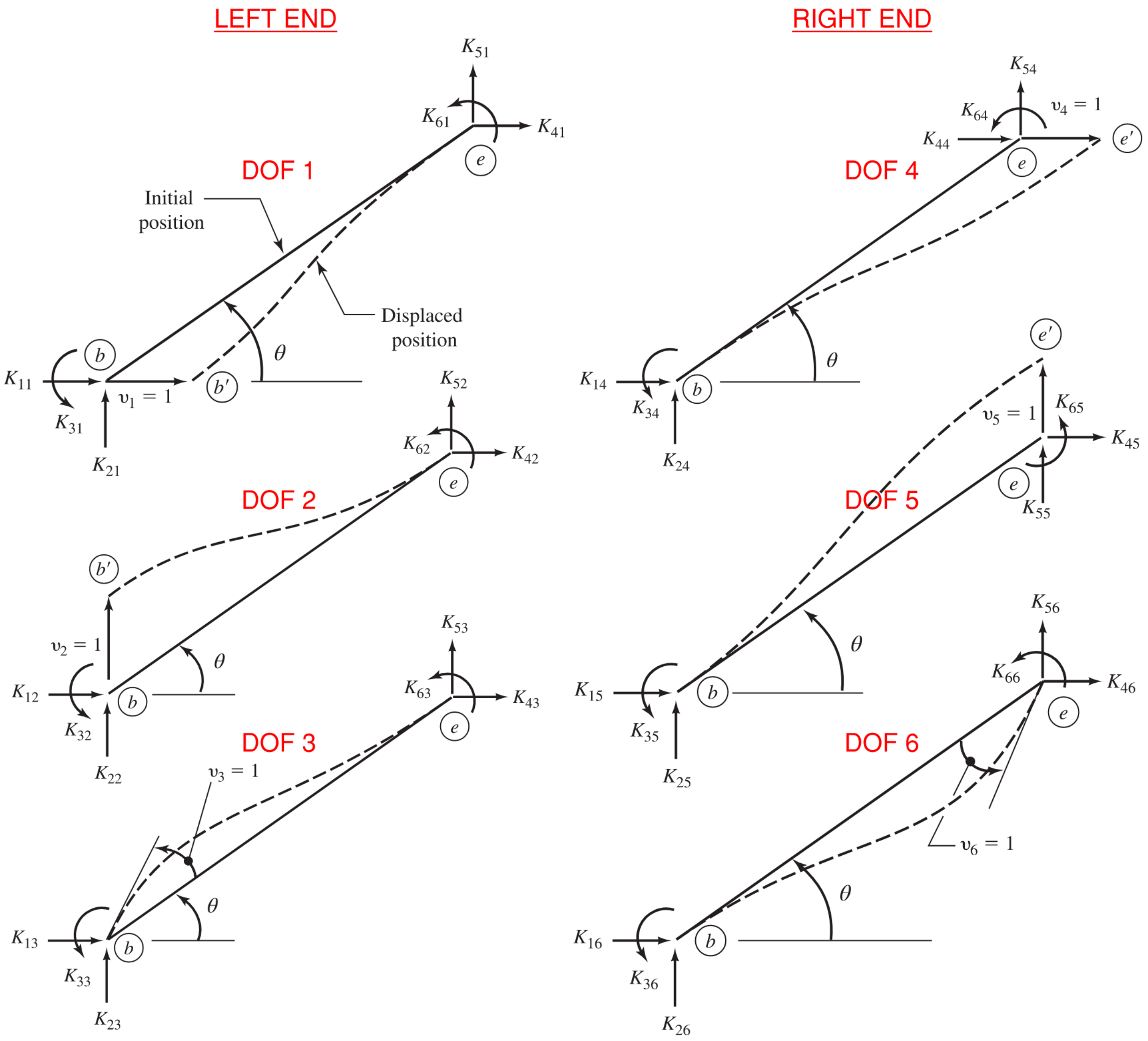
Part 3 — 2D Frame Element Stiffness Matrix

Global Coordinates

Unit Displacement Method

You can derive the 2D frame element from scratch using the unit displacement method:

$$k_{ij} = \text{force at DOF } i \text{ due to a unit displacement at DOF } j,$$
$$\text{with all other DOFs fixed.}$$



LEFT END
RIGHT END
DOF 1
DOF 2
DOF 3
DOF 4
DOF 5
DOF 6
Initial position
Displaced position
K_{11}
K_{21}
K_{31}
K_{41}
K_{51}
K_{61}
$v_1 = 1$
K_{12}
K_{22}
K_{32}
K_{42}
K_{52}
K_{62}
$v_2 = 1$
K_{13}
K_{23}
K_{33}
K_{43}
K_{53}
K_{63}
$v_3 = 1$
K_{14}
K_{24}
K_{34}
K_{44}
K_{54}
K_{64}
$v_4 = 1$
K_{15}
K_{25}
K_{35}
K_{45}
K_{55}
K_{65}
$v_5 = 1$
K_{16}
K_{26}
K_{36}
K_{46}
K_{56}
K_{66}
$v_6 = 1$
θ
b
b'
e
e'

Closed-Form Expression

A closed-form expression for the global member stiffness matrix **does exist** for frame elements (Kassimali, Section 6.4).

However, unlike the truss case, the resulting expression is more involved and not especially transparent.

$$
\mathbf{K} = \frac{EI}{L^3}\begin{bmatrix}
\frac{AL^2}{I}\cos^2\theta + 12\sin^2\theta & \left(\frac{AL^2}{I} - 12\right)\cos\theta\sin\theta & -6L\sin\theta & -\left(\frac{AL^2}{I}\cos^2\theta + 12\sin^2\theta\right) & -\left(\frac{AL^2}{I} - 12\right)\cos\theta\sin\theta & -6L\sin\theta \\
\left(\frac{AL^2}{I} - 12\right)\cos\theta\sin\theta & \frac{AL^2}{I}\sin^2\theta + 12\cos^2\theta & 6L\cos\theta & -\left(\frac{AL^2}{I} - 12\right)\cos\theta\sin\theta & -\left(\frac{AL^2}{I}\sin^2\theta + 12\cos^2\theta\right) & 6L\cos\theta \\
-6L\sin\theta & 6L\cos\theta & 4L^2 & 6L\sin\theta & -6L\cos\theta & 2L^2 \\
-\left(\frac{AL^2}{I}\cos^2\theta + 12\sin^2\theta\right) & -\left(\frac{AL^2}{I} - 12\right)\cos\theta\sin\theta & 6L\sin\theta & \frac{AL^2}{I}\cos^2\theta + 12\sin^2\theta & \left(\frac{AL^2}{I} - 12\right)\cos\theta\sin\theta & 6L\sin\theta \\
-\left(\frac{AL^2}{I} - 12\right)\cos\theta\sin\theta & -\left(\frac{AL^2}{I}\sin^2\theta + 12\cos^2\theta\right) & -6L\cos\theta & \left(\frac{AL^2}{I} - 12\right)\cos\theta\sin\theta & \frac{AL^2}{I}\sin^2\theta + 12\cos^2\theta & -6L\cos\theta \\
-6L\sin\theta & 6L\cos\theta & 2L^2 & 6L\sin\theta & -6L\cos\theta & 4L^2
\end{bmatrix}
$$

For this reason, it is generally cleaner and more systematic to compute:

$$\mathbf{K} = \mathbf{T}^{\mathsf{T}}\mathbf{k}\mathbf{T}$$

Part 4 — Member Forces

> *Global Coordinates*

Why Do We Rotate to Global?

We need fixed-end forces (FEFs) in **global coordinates** before assembly, because the global system is written and solved in the **global coordinate system**.

Mathematical Expression

The force transformation for fixed-end forces is:

$$\mathbf{F}^F = \mathbf{T}^\mathsf{T}\mathbf{Q}^F$$

Using the same $\mathbf{T}$ as before:

$$\mathbf{T}^\mathsf{T} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\theta & -\sin\theta & 0 \\ 0 & 0 & 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

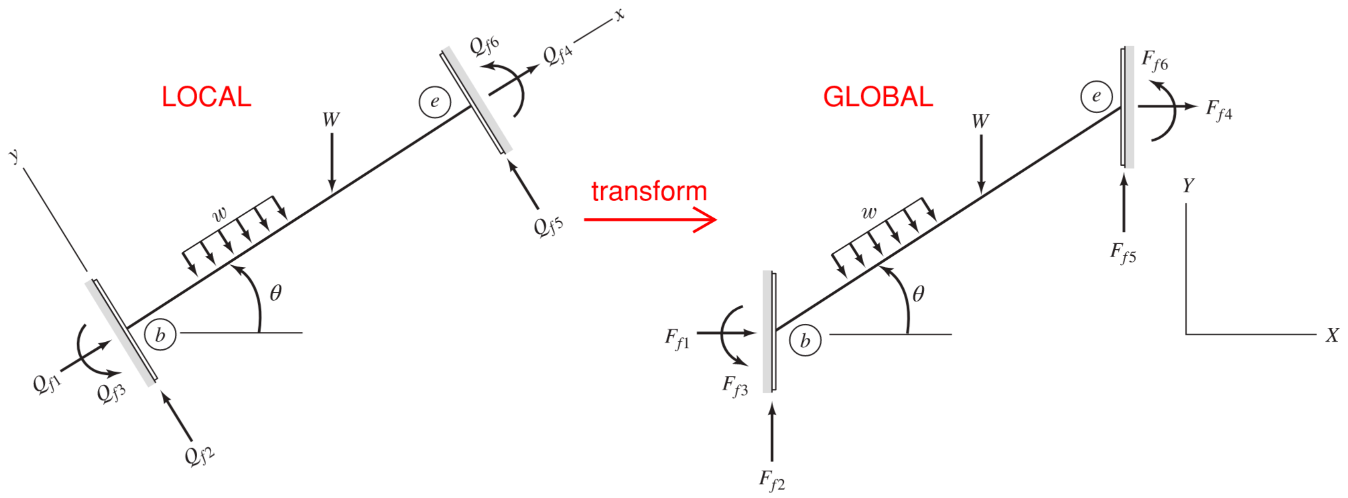
Expression for $\mathbf{F}^F$

Let $c = \cos\theta$ and $s = \sin\theta$. Then:

$$\begin{Bmatrix} F_1^F \\ F_2^F \\ F_3^F \\ F_4^F \\ F_5^F \\ F_6^F \end{Bmatrix} = \mathbf{T}^\mathsf{T} \begin{Bmatrix} Q_1^F \\ Q_2^F \\ Q_3^F \\ Q_4^F \\ Q_5^F \\ Q_6^F \end{Bmatrix} = \begin{Bmatrix} c\,Q_1^F - s\,Q_2^F \\ s\,Q_1^F + c\,Q_2^F \\ Q_3^F \\ c\,Q_4^F - s\,Q_5^F \\ s\,Q_4^F + c\,Q_5^F \\ Q_6^F \end{Bmatrix}$$

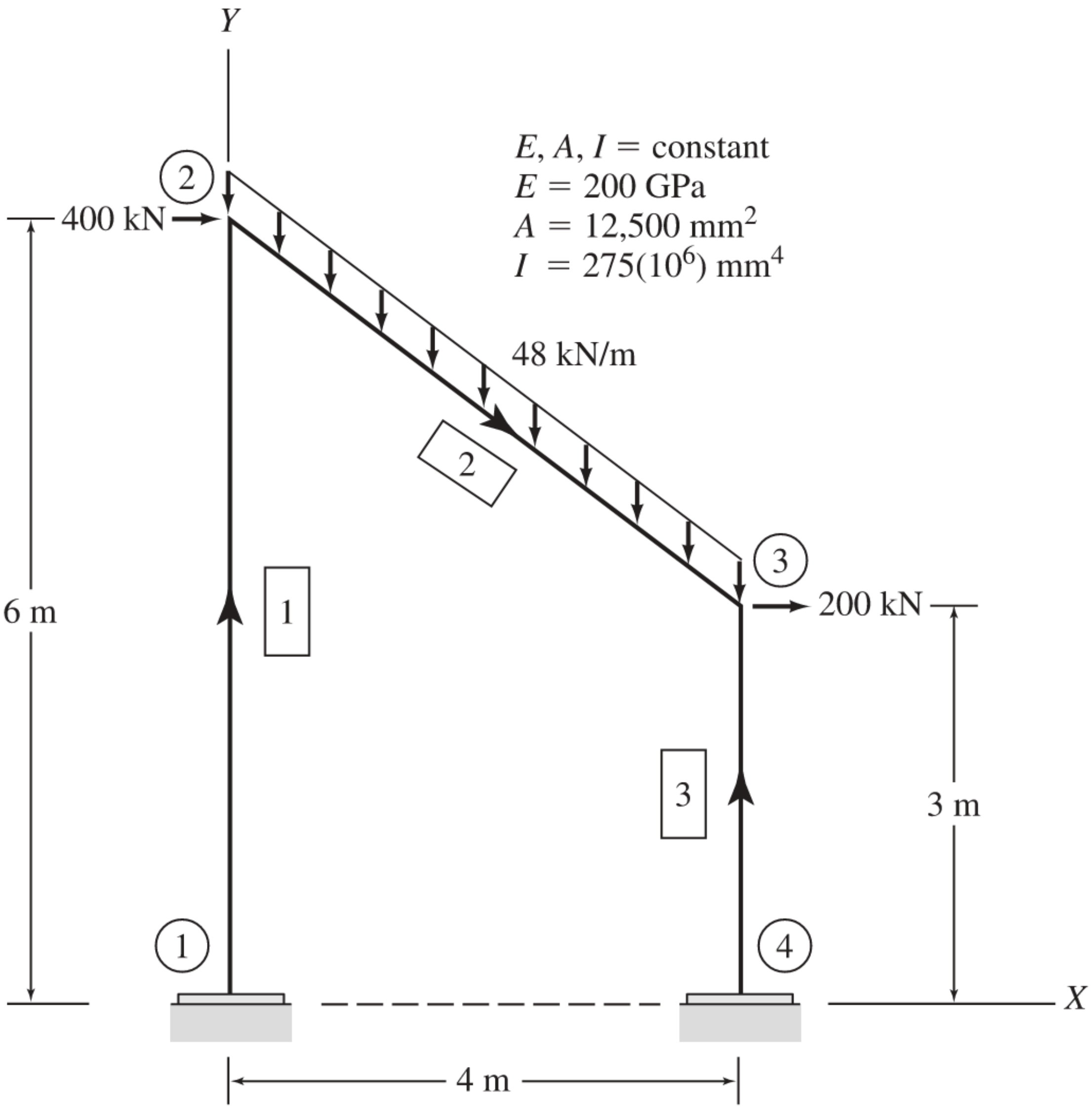
Transformation Process

1. Compute the member fixed-end force vector in **local** coordinates: $\mathbf{Q}^F$
2. Rotate it into **global** coordinates: $\mathbf{F}^F = \mathbf{T}^\mathsf{T}\mathbf{Q}^F$
3. Assemble $\mathbf{F}^F$ into the global load vector (same coordinates as the global stiffness matrix)



Part 5 — Example (Kassimali 6.3 and 6.4)

Continue with the same structure.

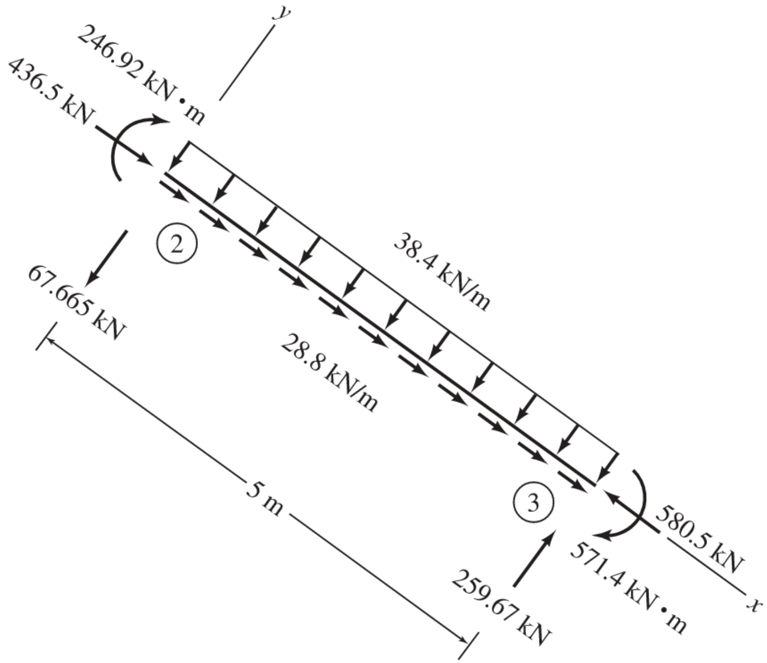


Local → Global Element Forces

If given local element forces from an earlier step:

$$\mathbf{F} = \mathbf{T}^{\mathsf{T}}\mathbf{Q}$$

$$\mathbf{Q} = \begin{bmatrix} 436.5 & -67.669 & -246.929 & -580.5 & 259.669 & -571.418 \end{bmatrix}^T$$



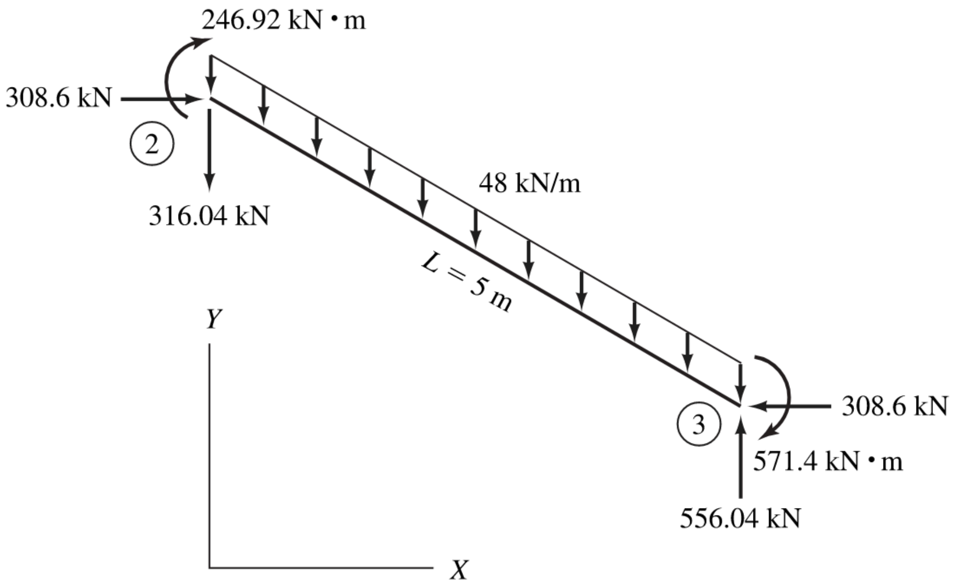
In [1]:
```python
import numpy as np
np.set_printoptions(precision=3, suppress=True)

def transformation_matrix(theta):
    """Return the 6x6 transformation matrix T for a 2D frame element (th
    theta = np.radians(theta)  # degrees → radians
    c = np.cos(theta)
    s = np.sin(theta)

    T = np.array([
        [ c,   s, 0, 0, 0, 0],
        [-s,   c, 0, 0, 0, 0],
        [ 0,   0, 1, 0, 0, 0],
        [ 0,   0, 0, c, s, 0],
        [ 0,   0, 0, -s, c, 0],
        [ 0,   0, 0, 0, 0, 1]
    ], dtype=float)

    return T
```

In [2]:
```python
theta = 270 + np.degrees(np.arctan(4/3))

T = transformation_matrix(theta)
print(T)
```

```
[[ 0.8 -0.6  0.   0.   0.   0. ]
 [ 0.6  0.8  0.   0.   0.   0. ]
 [ 0.   0.   1.   0.   0.   0. ]
 [ 0.   0.   0.   0.8 -0.6  0. ]
 [ 0.   0.   0.   0.6  0.8  0. ]
 [ 0.   0.   0.   0.   0.   1. ]]
```

In [3]:
```
Q = np.array([436.5, -67.669, -246.929, -580.5, 259.669, -571.418])

F = T.T @ Q
print(F)
```

```
[ 308.599 -316.035 -246.929 -308.599  556.035 -571.418]
```

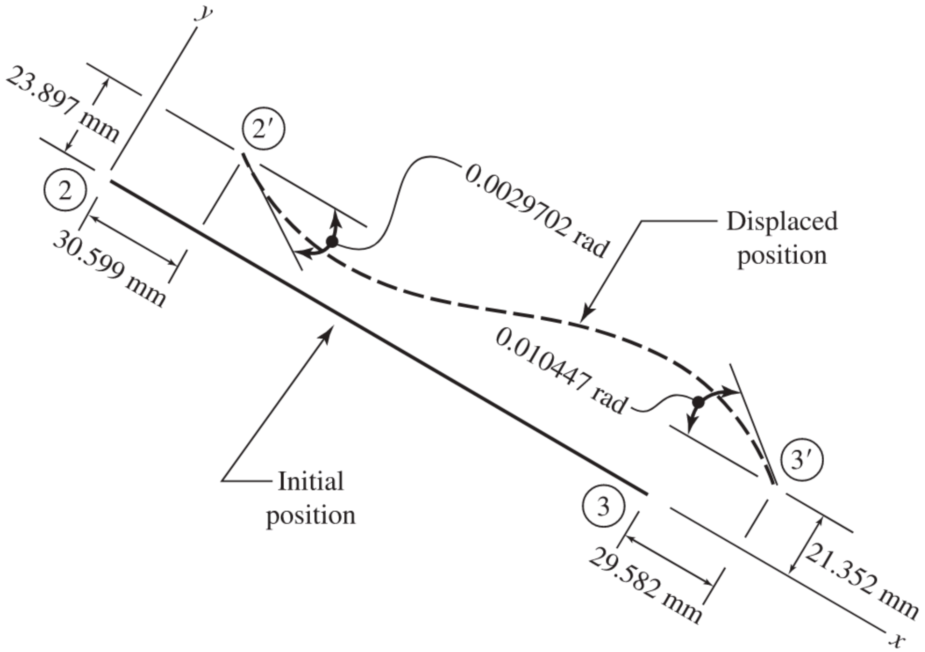


Local → Global Element Displacements

If given local element displacements from an earlier step:

$$\mathbf{v} = \mathbf{T}^{\mathsf{T}}\mathbf{u}$$

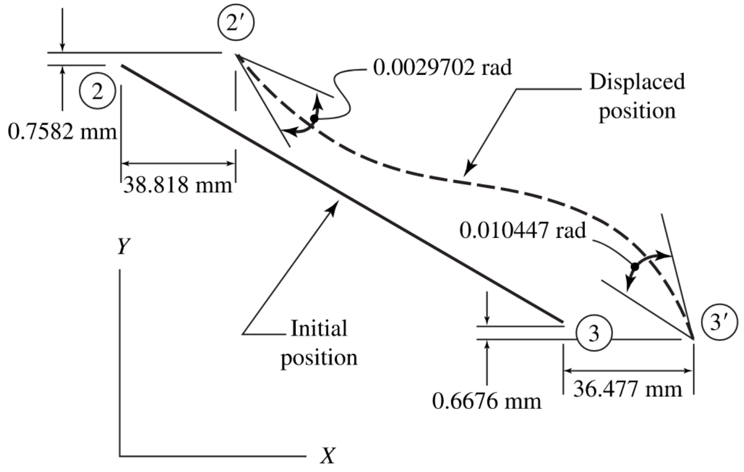
$$\mathbf{u} = \begin{bmatrix} 0.030599 & 0.023897 & -0.0029702 & 0.029582 & 0.021352 & -0.010447 \end{bmatrix}^{T}$$



In [4]:

```python
# displacement vector (units: m, rad)
u = np.array([
    0.030599,
    0.023897,
   -0.0029702,  # clockwise rotation is negative
    0.029582,
    0.021352,
   -0.010447    # clockwise rotation is negative
], dtype=float)

v = T.T @ u
print(v)
```

```
[ 0.039  0.001 -0.003  0.036 -0.001 -0.01 ]
```



Element Forces Using the Global Stiffness Relationship

$$\mathbf{F} = \mathbf{T}^{\mathsf{T}}\mathbf{k}\mathbf{T}\mathbf{v} + \mathbf{T}^{\mathsf{T}}\mathbf{Q}^{F}$$

Given:

$$\mathbf{k} = \begin{bmatrix} 500000 & 0 & 0 & -500000 & 0 & 0 \\ 0 & 5280 & 13200 & 0 & -5280 & 13200 \\ 0 & 13200 & 44000 & 0 & -13200 & 22000 \\ -500000 & 0 & 0 & 500000 & 0 & 0 \\ 0 & -5280 & -13200 & 0 & 5280 & -13200 \\ 0 & 13200 & 22000 & 0 & -13200 & 44000 \end{bmatrix}$$

$$\mathbf{Q}^{F} = \begin{bmatrix} -72 & 96 & 80 & -72 & 96 & -80 \end{bmatrix}^{T}$$

$$\mathbf{v}$$

$$= \begin{bmatrix} 0.0388174 & 0.0007582 & -0.0029702 & 0.0364768 & -0.0006676 & -0.010447 \end{bmatrix}^{T}$$

In [5]:

```python
k = np.array([
    [ 500000,      0,      0, -500000,      0,      0],
    [      0,   5280, 13200,       0, -5280, 13200],
    [      0, 13200, 44000,       0, -13200, 22000],
    [-500000,      0,      0,  500000,      0,      0],
    [      0, -5280, -13200,       0,  5280, -13200],
    [      0, 13200, 22000,       0, -13200, 44000]
], dtype=float)

K = T.T @ k @ T
print(K)
```

```
[[ 321900.8 -237465.6    7920.   -321900.8  237465.6    7920. ]
 [-237465.6  183379.2   10560.    237465.6 -183379.2   10560. ]
 [   7920.     10560.   44000.     -7920.    -10560.   22000. ]
 [-321900.8  237465.6   -7920.    321900.8 -237465.6   -7920. ]
 [ 237465.6 -183379.2  -10560.   -237465.6  183379.2  -10560. ]
 [   7920.     10560.   22000.     -7920.    -10560.   44000. ]]
```

In [6]:
```
Qf = np.array([-72, 96, 80, -72, 96, -80], dtype=float)

Ff = T.T @ Qf
print(Ff)
```

```
[   0. 120.   80.    0. 120. -80.]
```

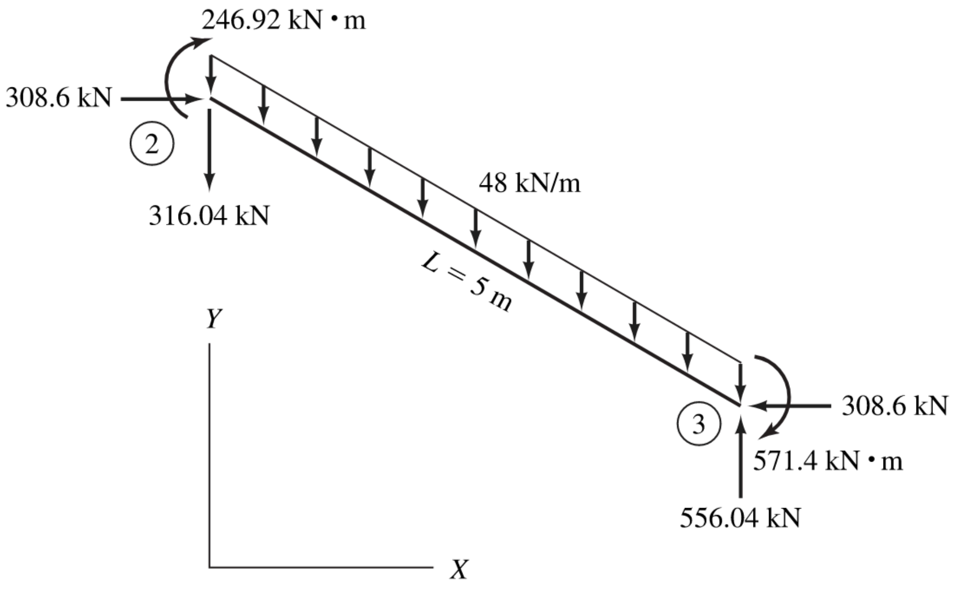
In [7]:

```python
# Should give the same result as T.T @ Qf

def transform_Qf_to_global(Qf, theta):
    """Compute F^F = T^T Q^F using the closed-form expressions (theta ir
    theta = np.radians(theta)
    c = np.cos(theta)
    s = np.sin(theta)

    Q1, Q2, Q3, Q4, Q5, Q6 = Qf

    Ff = np.array([
        c*Q1 - s*Q2,
        s*Q1 + c*Q2,
        Q3,
        c*Q4 - s*Q5,
        s*Q4 + c*Q5,
        Q6
    ], dtype=float)

    return Ff

Ff = transform_Qf_to_global(Qf, theta)
print(Ff)
```

```
[  0. 120.  80.   0. 120. -80.]
```

In [8]:
```
v = np.array([0.0388174, 0.0007582, -0.0029702,
              0.0364768, -0.0006676, -0.010447], dtype=float)

F = K @ v + Ff
print(F)
```

```
[ 308.598 -316.036 -246.929 -308.598  556.036 -571.418]
```



Part 6 — DSM Full Procedure for 2D Frames

Forward Pass — Global Structural Analysis

1. **Define the structure**
    - Node numbering and coordinates
    - Global DOF numbering
    - Element connectivity
    - Support restraints and applied loads
2. **Compute element stiffness matrices**
    - For each element, compute geometry: L, θ
    - Build the transformation matrix $\boldsymbol{T}$
    - Compute global element stiffness:

$$\boldsymbol{k} = \boldsymbol{T}^\top \boldsymbol{k}' \boldsymbol{T}$$

3. **Compute element fixed-end force (FEF) vectors (if needed)**

   - Identify the load case
   - Compute local closed-form FEFs, $\boldsymbol{Q}^F$
   - Transform to global:

$$\boldsymbol{F}^F = \boldsymbol{T}^\top \boldsymbol{Q}^F$$

4. **Assemble global stiffness matrix**

   - Scatter-add element stiffness contributions into $\boldsymbol{K}$

5. **Assemble global load vectors**

   - Assemble element FEF contributions into $\boldsymbol{F}^F$
   - Assemble applied joint loads into $\boldsymbol{F}$

6. **Apply boundary conditions**
   - Partition DOFs into free (f) and restrained (r)
7. **Solve for unknown displacements**

$$\boldsymbol{u}_f = \boldsymbol{K}_{ff}^{-1}\left(\boldsymbol{F}_f - \boldsymbol{F}_f^F - \boldsymbol{K}_{fr}\boldsymbol{u}_r\right)$$

8. **Recover support reactions**

$$\boldsymbol{F}_r = \boldsymbol{K}_{rf}\boldsymbol{u}_f + \boldsymbol{K}_{rr}\boldsymbol{u}_r + \boldsymbol{F}_r^F$$

Backward Pass — Element Recovery and Design

1. **Extract element global displacement vectors**
    - For each member, collect the relevant entries from $\boldsymbol{u}$ to form $\boldsymbol{u}_{(e)}$
2. **Transform displacements to local coordinates**

$$\boldsymbol{u}' = \boldsymbol{T}\,\boldsymbol{u}_{(e)}$$

3. **Extract element global FEF vectors**
    - For each member, collect the relevant entries from $\boldsymbol{F}^F$ to form $\boldsymbol{F}^F_{(e)}$
4. **Transform FEFs to local coordinates**

$$\boldsymbol{Q}^F_{(e)} = \boldsymbol{T}\,\boldsymbol{F}^F_{(e)}$$

5. **Compute local element end forces and moments**

$$\boldsymbol{f}' = \boldsymbol{k}'\,\boldsymbol{u}' + \boldsymbol{Q}^F$$

6. **Compute design quantities**

- Use local end forces to draw shear and bending moment diagrams (per member)
- Compute axial and bending stresses as needed

Wrap-Up

In this lecture, we:

- Reviewed local ↔ global transformations for forces and displacements
- Applied transformtation in a worked example
- Built the global member relation $\mathbf{F} = \mathbf{K}\mathbf{u} + \mathbf{F}^F$
- Outlined the DSM procedure