CEE6501 — Lecture 3.2

The Direct Stiffness Method (DSM) for Trusses

Learning Objectives

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

- Build a local-to-global transformation for a truss member
- Compute element stiffness in global coordinates: $[k]_g = [T]^T[k'][T]$
- Assemble the global stiffness matrix $[K]$ by scatter-add
- Explain why an unsupported structure yields a singular stiffness matrix
- Apply boundary conditions via partitioning and solve for displacements
- Recover member axial forces from global displacements

Agenda

Part 2 (today): Global behavior and the Direct Stiffness Method

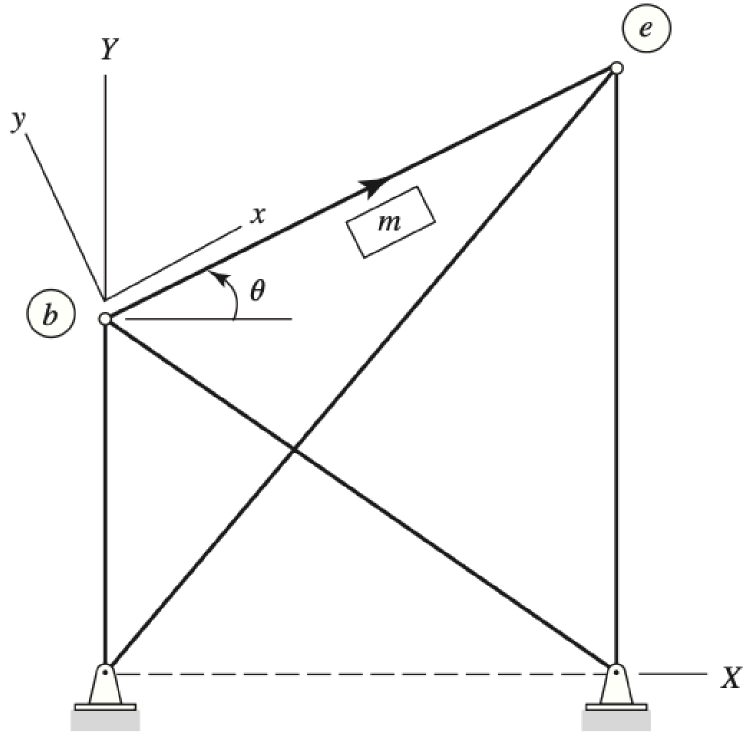
1. Transformation from local to global (direction cosines)
2. Member stiffness relations in the global coordinate system
3. Nodal equilibrium and why assembly works
4. Assemble the global stiffness matrix $[K]$
5. First attempt: no supports (singular $[K]$)
6. Constraints and supports
7. Partitioning into free vs restrained DOFs
8. Solve for global displacements
9. Second attempt: with supports
10. Recover element forces in local coordinates
11. DSM summary: step-by-step procedure
12. Stiffness matrix features + indeterminacy

Big idea:

- A truss is a network of axial springs.
- Each element contributes stiffness to shared DOFs.
- Assembly is adding contributions into the right global rows/columns.

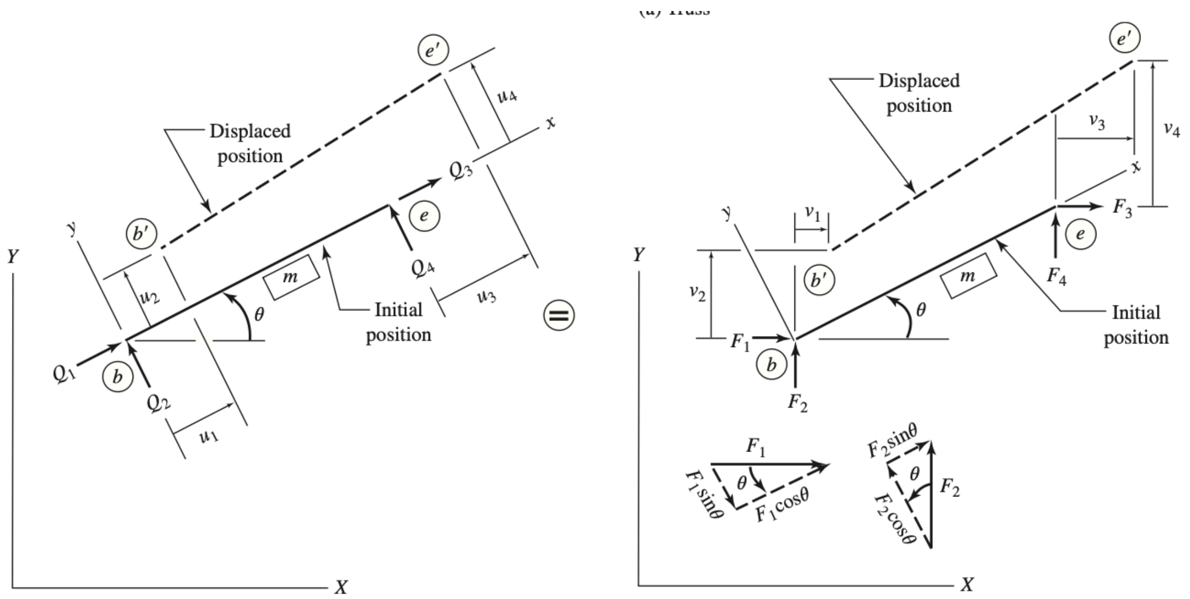
Part 1 — Local to Global Transformation

Truss Element in a Structure



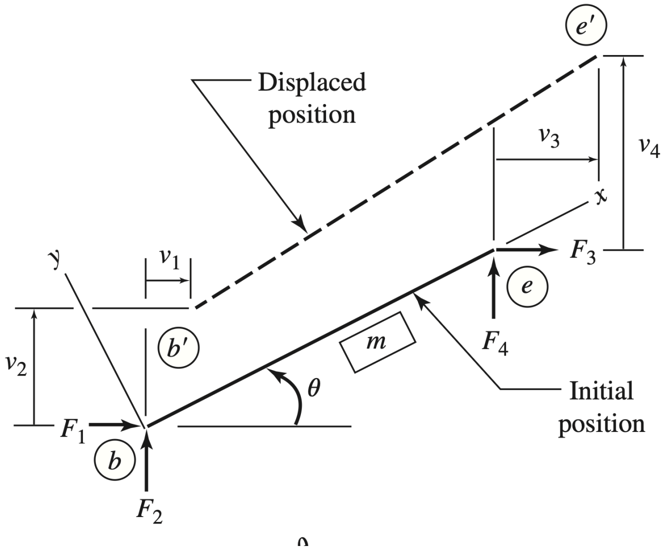
- A truss member is embedded in a **global coordinate system** (X, Y)
- The element stiffness was derived in a **local coordinate system** (x, y) aligned with the member
- The member orientation is defined by an angle θ, measured **counterclockwise** from global X to local x
- Structural assembly requires **transforming forces and displacements** between local and global coordinates

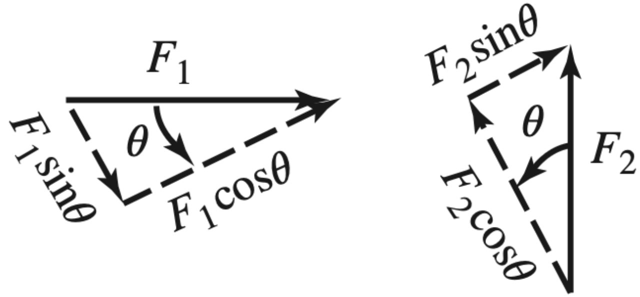
Transformation Perspectives



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

Global → Local Forces (Trigonometry)





At node b (start node):

- $Q_1 = F_1 \cos\theta + F_2 \sin\theta$
- $Q_2 = -F_1 \sin\theta + F_2 \cos\theta$

At node e (end node):

- $Q_3 = F_3 \cos\theta + F_4 \sin\theta$
- $Q_4 = -F_3 \sin\theta + F_4 \cos\theta$

Global → Local Force Transformation (Matrix Form)

- Local member forces $\boldsymbol{Q}$ are obtained by **rotating** global nodal forces $\boldsymbol{F}$ into the member's local coordinate system
- Each 2×2 block applies a **rotation by** θ at a node
- The transformation changes **direction only**, not force magnitude
- This operation is a **pure coordinate transformation**

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

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

Direction Cosines (Rotation Terms)

- Direction cosines define the **orientation of a truss member** in the global (X, Y) coordinate system
- The angle θ is measured **counterclockwise** from the global X axis to the local x axis
- Computed directly from the **nodal coordinates** of the element (b = start node, e = end node)

$$\cos\theta = \frac{X_e - X_b}{\sqrt{(X_e - X_b)^2 + (Y_e - Y_b)^2}}, \qquad \sin\theta = \frac{Y_e - Y_b}{\sqrt{(X_e - X_b)^2 + (Y_e - Y_b)^2}}$$

- The denominator is the **member length** L
- Once computed, $\cos\theta$ and $\sin\theta$ are **reused throughout the element formulation**

Global → Local Displacements

- Nodal displacements are transformed using the **same rotation matrix** as forces
- Displacements and forces transform identically because they are defined along the **same directions**
- This is a **pure coordinate rotation**, not a change in deformation

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

- $\boldsymbol{v}$: global displacement vector
- $\boldsymbol{u}$: local displacement vector

Local → Global Force Transformation

- This is the **reverse of the global → local process**
- Local member forces are **rotated back** into the global (X, Y) directions

At node b (start node):

$$F_1 = Q_1 \cos\theta - Q_2 \sin\theta, \qquad F_2 = Q_1 \sin\theta + Q_2 \cos\theta$$

At node e (end node):

$$F_3 = Q_3 \cos\theta - Q_4 \sin\theta, \qquad F_4 = Q_3 \sin\theta + Q_4 \cos\theta$$

Local ⟶ Global Force Transformation (Matrix Form)

- Local member forces are mapped to global nodal forces using the **transpose** of the global→local transformation

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{Bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & \cos\theta & -\sin\theta \\ 0 & 0 & \sin\theta & \cos\theta \end{bmatrix}}_{\boldsymbol{T}^\top} \begin{Bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{Bmatrix}$$

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

Recall (Global → Local):

$$\boldsymbol{T} = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & \cos\theta & \sin\theta \\ 0 & 0 & -\sin\theta & \cos\theta \end{bmatrix}$$

Properties of the Transformation Matrix

$$\boldsymbol{T}^{-1} = \boldsymbol{T}^T$$

The transformation matrix is **orthogonal**, which greatly simplifies operations involving the stiffness transformations.

Summary — Local ↔ Global Transformations

Direction cosines (member orientation):

$$\cos\theta = \frac{X_e - X_b}{L}, \qquad \sin\theta = \frac{Y_e - Y_b}{L}$$

Global → Local (forces or displacements):

$$\boldsymbol{Q} = \boldsymbol{T}\boldsymbol{F}, \qquad \boldsymbol{u} = \boldsymbol{T}\boldsymbol{v}$$

Local → Global (forces or displacements):

$$\boldsymbol{F} = \boldsymbol{T}^{\mathsf{T}}\boldsymbol{Q}, \qquad \boldsymbol{v} = \boldsymbol{T}^{\mathsf{T}}\boldsymbol{u}$$

$$\boldsymbol{T} = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & \cos\theta & \sin\theta \\ 0 & 0 & -\sin\theta & \cos\theta \end{bmatrix}$$

- $\boldsymbol{T}$ is a **pure rotation matrix**
- $\boldsymbol{T}^{-1} = \boldsymbol{T}^{\mathsf{T}}$ (orthogonal)

Part 2 — Member Stiffness in the Global Coordinate System

Goal

- We have derived the **local stiffness relation** (Lecture 3.1 today):

$$\boldsymbol{Q} = \boldsymbol{k}\boldsymbol{u}$$

- We also know how to **transform forces and displacements** between local and global systems
- Objective: express the **member stiffness relation entirely in global coordinates**

Transformation Chain — From Local to Global (Step-by-Step)

Step 1 — Local force–displacement relation

$$\boldsymbol{Q} = \boldsymbol{k}\,\boldsymbol{u}$$

The element stiffness matrix $\boldsymbol{k}$ relates **local nodal displacements** $\boldsymbol{u}$ to the corresponding **local nodal forces** $\boldsymbol{Q}$.

Step 2 — Transform local forces to global forces

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

Global nodal forces are obtained by rotating the local force vector into the global coordinate system.

Substitute the local stiffness relation from Step 1, $\boldsymbol{Q} = \boldsymbol{k}\,\boldsymbol{u}$, into the force transformation:

$$\boldsymbol{F} = \boldsymbol{T}^{\top}\boldsymbol{Q} \implies \boldsymbol{F} = \boldsymbol{T}^{\top}(\boldsymbol{k}\,\boldsymbol{u})$$

Step 3 — Transform global displacements to local displacements

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

Substitute the displacement transformation into the previous expression:

$$\boldsymbol{F} = \boldsymbol{T}^{\mathsf{T}}\boldsymbol{k}\,\boldsymbol{u} \implies \boldsymbol{F} = \boldsymbol{T}^{\mathsf{T}}\boldsymbol{k}\big(\boldsymbol{T}\,\boldsymbol{v}\big)$$

Step 4 — Rearrange into global stiffness form

$$\boldsymbol{F} = \left(\boldsymbol{T}^{\mathsf{T}}\,\boldsymbol{k}\,\boldsymbol{T}\right)\boldsymbol{v}$$

Step 5 — Final global stiffness relation

Define the global element stiffness matrix:

$$\boldsymbol{K} = \boldsymbol{T}^{\mathsf{T}}\, \boldsymbol{k}\, \boldsymbol{T}$$

This is the element stiffness relation used directly in global assembly for the Direct Stiffness Method.

textbook notation:

$$\boxed{\boldsymbol{F} = \boldsymbol{K}\, \boldsymbol{v}}$$

our notation (interchangeable):

$$\boxed{\boldsymbol{f} = \boldsymbol{K}\, \boldsymbol{u}}$$

Calculating the Global Stiffness Matrix, $\boldsymbol{K}$

The global element stiffness matrix is obtained by **rotating the local axial stiffness** into the global coordinate system:

$$\boldsymbol{K} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & \cos\theta & -\sin\theta \\ 0 & 0 & \sin\theta & \cos\theta \end{bmatrix} \cdot \frac{EA}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & \cos\theta & \sin\theta \\ 0 & 0 & -\sin\theta & \cos\theta \end{bmatrix}$$

Carrying out the matrix multiplication yields the **closed-form global stiffness matrix**:

$$\boldsymbol{K} = \frac{EA}{L} \begin{bmatrix} \cos^2\theta & \cos\theta\sin\theta & -\cos^2\theta & -\cos\theta\sin\theta \\ \cos\theta\sin\theta & \sin^2\theta & -\cos\theta\sin\theta & -\sin^2\theta \\ -\cos^2\theta & -\cos\theta\sin\theta & \cos^2\theta & \cos\theta\sin\theta \\ -\cos\theta\sin\theta & -\sin^2\theta & \cos\theta\sin\theta & \sin^2\theta \end{bmatrix}$$

Key Observations — Global Stiffness Matrix, $\boldsymbol{K}$

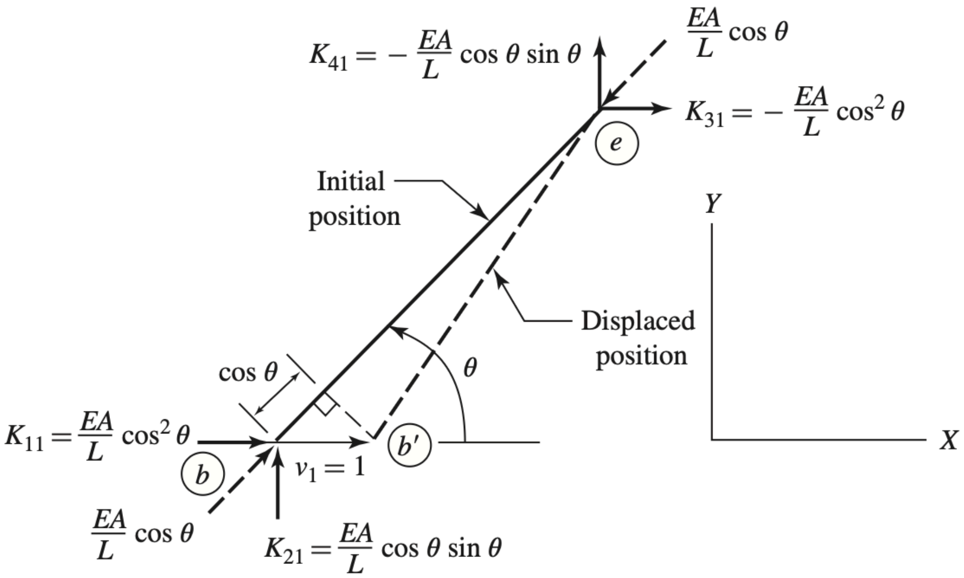
- The member global stiffness matrix $\boldsymbol{K}$ is **symmetric**, just like the local stiffness matrix
- $\boldsymbol{K}$ represents the same physical behavior, but expressed in the **global** (X, Y) **coordinate system**
- Each coefficient K_{ij} is the **force at global DOF** i required to produce a **unit displacement at global DOF** j, with all other displacements fixed

Direct Calculation of Global Stiffness Matrix, $\boldsymbol{K}$

- One *could* derive $\boldsymbol{K}$ directly by:
    - Applying **unit global displacements** to a generic inclined truss member
    - Evaluating the **global end forces** required to produce each unit displacement in global coordinates
- The j**th column of $\boldsymbol{K}$** gives the global nodal force pattern caused by $v_j = 1$
- This approach is **theoretically equivalent** to the transformation-based derivation and provides a clear physical interpretation of $\boldsymbol{K}$.
- However, it is **significantly more labor-intensive**, and is mainly useful as a **verification tool**, rather than for routine analysis.

Example: First Column of $\boldsymbol{K}$

$$u_a = v_1 \cos\theta = 1\cos\theta = \cos\theta$$



(a) First Column of **K** ($v_1 = 1, v_2 = v_3 = v_4 = 0$)

- Impose $v_1 = 1$, all other global displacements zero
- Project the resulting axial deformation onto the member axis
- Resolve the axial force back into global components

You recover the **first column of $\boldsymbol{K}$**, which should exactly match $\boldsymbol{T}^\top \boldsymbol{k}\boldsymbol{T}$.

Part 3 — Nodal Equilibrium and Why Assembly Works

> ***Draft note (delete later):*** *show equilibrium at a node, as a function of individual element stiffnesses, this is kind of a precursor to doing a direct assembly*

At a joint, equilibrium requires:

- Sum of element end forces (in global components) + external load = 0

Matrix viewpoint:

- Each element contributes forces into a shared pool of DOFs.
- Assembly collects these contributions into a global equation:

$$\{F\} = [K]\{u\}$$

Part 2.4 — Assembling the Global Stiffness Matrix

> ***Draft note (delete later):*** *assembling the global stiffness matrix, from local matrices rotated to global, figure 3.15 Kassimalu really good*

Steps:

1. Choose a DOF numbering for all nodes
2. For each member:
    - compute $[k]_g$
    - add its terms into $[K]$ using the member's DOF indices

Implementation mindset:

- Start with $[K] = 0$
- For each element, scatter-add $[k]_g$ into $[K]$

Part 5 — Worked Example: First Attempt (No Supports)

> ***Draft note (delete later):*** *worked out solution (maybe example 3.1 McGuire, example no joints), also example 3.7 Kassimali but ignore constrained DOFs*

Goal:

- Assemble $[K]$ and $\{F\}$ for a small truss.
- Attempt to solve $\{F\} = [K]\{u\}$.

Observation:

- Without supports, $[K]$ is singular.

Part 6 — Constraints and Supports (Why $[K]$ Was Singular)

> ***Draft note (delete later):*** *constraints and supports, show the "instability" with no supports as an example (from last section), singular matrix etc, have to add constraints, how? dont actually need to worry, just add them in as normal and later you deal with --> partitioning*

Boundary conditions specify known displacements:

- e.g., a pin support might enforce $u_x = 0$ and $u_y = 0$
- a roller might enforce one component only

Physical meaning:

- Rigid-body modes exist without constraints
- Some displacement patterns produce no strain energy

Part 7 — Partitioning the Matrix (Free vs Restrained)

> ***Draft note (delete later):*** *partitioning the matrix, into fixed and free*

Reorder DOFs into:

- free DOFs: f
- restrained DOFs: r

$$\begin{bmatrix} K_{ff} & K_{fr} \\ K_{rf} & K_{rr} \end{bmatrix} \begin{bmatrix} u_f \\ u_r \end{bmatrix} = \begin{bmatrix} F_f \\ F_r \end{bmatrix}$$

Part 8 — Solving for Global Displacements

> ***Draft note (delete later):*** *solving everythign, finding global displacements, matrix algebra here, with partitioned matrix, page 41 of McGuire shows this well*

Typically u_r is known (often zeros).

Solve the free subsystem:

$$u_f = K_{ff}^{-1}(F_f - K_{fr}u_r)$$

Then recover reactions:

$$F_r = K_{rf}u_f + K_{rr}u_r$$

Part 9 — Worked Example: Second Attempt (With Supports)

> ***Draft note (delete later):*** *now solve McGuire 3.2, same structure as 3.1 but now with supports*

Now apply supports and solve for:

- global displacements $\{u\}$
- reactions at restrained DOFs

Part 10 — Recover Element Forces (Back to Local)

> ***Draft note (delete later):*** *now go back to local, for element forces etc. Section 3.8 Kassimalu explains, in the step by step guide*

For each member:

1. Extract element global displacement vector $\{u\}_e$
2. Transform to local: $\{u'\} = [T]\{u\}_e$
3. Compute local end forces: $\{f'\} = [k']\{u'\}$
4. Axial force: $N = \frac{EA}{L}(u'_2 - u'_1)$

Part 11 — DSM Summary (Explicit Steps)

> ***Draft note (delete later):*** *the direct stiffness method summary (what we just did), steps to solve, explicit steps*

1. Define geometry (nodes, members)
2. Number DOFs
3. For each member:
   - compute $L,\ \theta,\ c,\ s$
   - build $[T]$
   - compute $[k]_g = [T]^T[k'][T]$
4. Assemble $[K]$
5. Apply boundary conditions (partition into f, r)
6. Solve for u_f
7. Recover reactions F_r
8. Recover member forces/stresses in local coordinates

Part 12 — Features of the Stiffness Matrix + Indeterminacy

> ***Draft note (delete later):*** *some features of the stiffness matrix + indeterminacy (mcGuire 3.3, 3.4)*

Typical properties (for stable, properly constrained trusses):

- Symmetric
- Sparse
- Positive definite on free DOFs

Indeterminacy (conceptual):

- More members than needed for determinacy: stiffness method still works
- The matrix system enforces compatibility and equilibrium automatically

In [6]:
```
# (Optional) DSM code scaffold stub
# Placeholder for the hands-on notebook (or Lecture 3 lab section).

import numpy as np

# TODO: define nodes, connectivity, E, A
nodes = None          # e.g., np.array([[x1,y1],[x2,y2],...])
members = None        # e.g., list of (n1, n2)
E = None
A = None

# TODO: DOF numbering, element k_g, assembly, partitioning, solve, recov
K = None
F = None
u = None

K, F, u
```

Out[6]:
```
(None, None, None)
```

Wrap-Up

Today you built the DSM pipeline for trusses:

- local bar stiffness $\rightarrow$ transformation $\rightarrow$ global element stiffness
- assembly $\rightarrow$ constraints $\rightarrow$ solve $\rightarrow$ member force recovery

Next: implement DSM in Python for a worked truss example and discuss efficiency (sparsity/bandedness).