CEE6501 — Lecture 4.1

The Direct Stiffness Method (DSM) for Trusses (Part 2)

Learning Objectives

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

- Assemble the **global stiffness matrix K** using scatter-add
- Apply boundary conditions through **DOF partitioning** and solve for displacements
- Recover **member axial forces** from global displacements for design

Agenda

1. Review of the manual method for assembling the global stiffness matrix
2. Assembly of the global stiffness matrix K using the Direct Stiffness Method (DSM)
3. Constraints and supports (boundary conditions)
4. Partitioning into free and restrained DOFs
5. Solving for global displacements and reactions
6. Element force recovery
7. DSM summary

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 — Assembling the Global Structure Stiffness Matrix
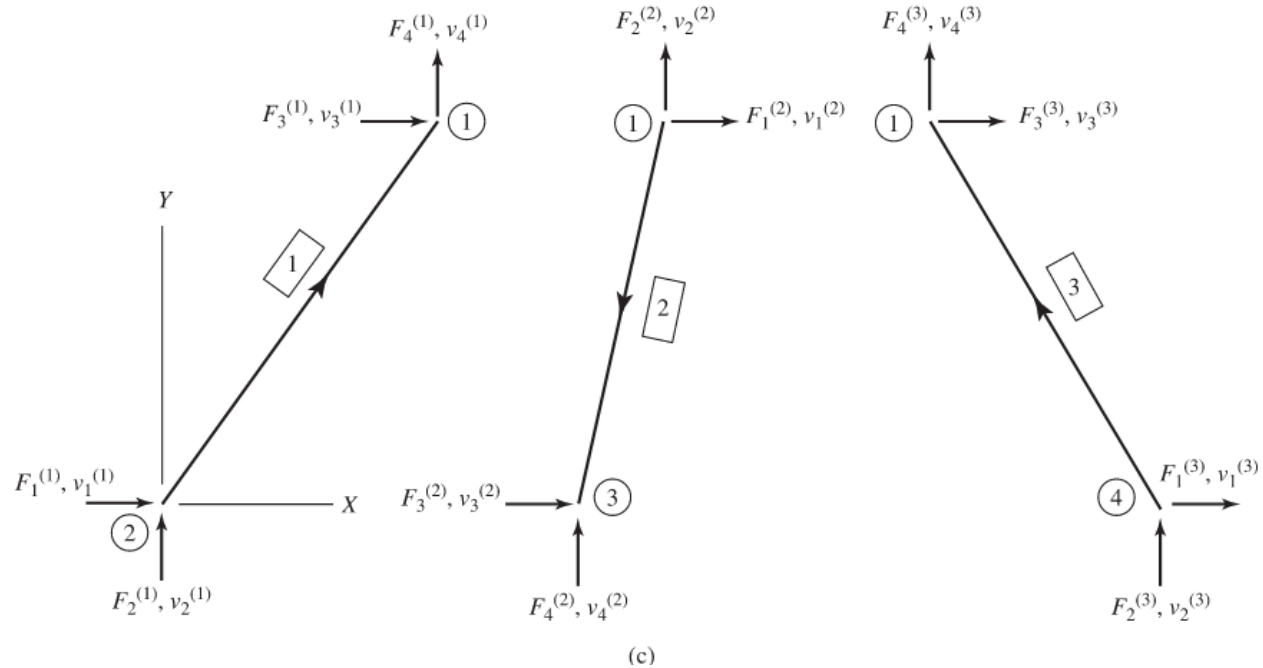
Review what we did last lecture

Example Structure

- Structure composed of **3 axial truss elements**
- **4 nodes**, each with (X, Y) translational DOFs, node 1 at the top
- Total system size: **8 global degrees of freedom**
  - node 1: DOF (1,2)
  - node 2: DOF (3,4)
  - node 3: DOF (5,6)
  - node 4: DOF (7,8)

No description has been provided for this image
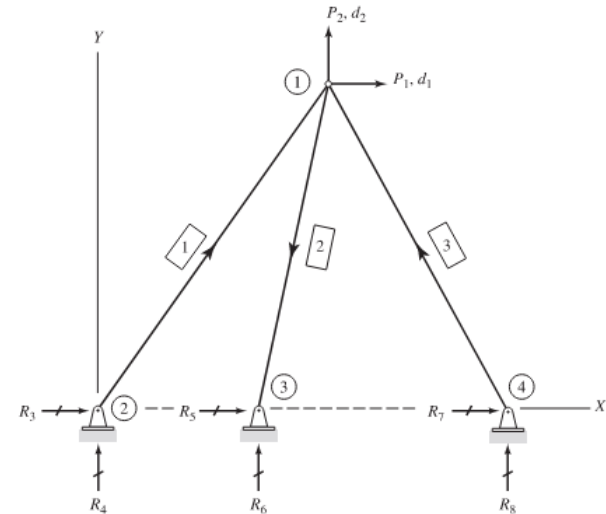
Element-Level View of the Structure

$$P_1 = F_3^{(1)} + F_1^{(2)} + F_3^{(3)}$$
$$P_2 = F_4^{(1)} + F_2^{(2)} + F_4^{(3)}$$

Summary of Force–Displacement Relations at Free Node

We derived these equations using displacement compatibility coupled with member-level equilbrium (see last lecture for more details on this process)

The forces acting at **end node of member 1** (Global node 1) are:

$$F_3^{(1)} = K_{33}^{(1)} d_1 + K_{34}^{(1)} d_2$$

$$F_4^{(1)} = K_{43}^{(1)} d_1 + K_{44}^{(1)} d_2$$

The forces acting at **start node of member 2** (Global node 1) are:

$$F_1^{(2)} = K_{11}^{(2)} d_1 + K_{12}^{(2)} d_2$$

$$F_2^{(2)} = K_{21}^{(2)} d_1 + K_{22}^{(2)} d_2$$

The forces acting at **end node of member 3** (Global node 1) are:

$$F_3^{(3)} = K_{33}^{(3)} d_1 + K_{34}^{(3)} d_2$$

$$F_4^{(3)} = K_{43}^{(3)} d_1 + K_{44}^{(3)} d_2$$

Global Stiffness Matrix for Free DOFs, $\boldsymbol{K_{ff}}$

$$P_1 = \left(K_{33}^{(1)} + K_{11}^{(2)} + K_{33}^{(3)}\right)d_1 + \left(K_{34}^{(1)} + K_{12}^{(2)} + K_{34}^{(3)}\right)d_2$$

$$P_2 = \left(K_{43}^{(1)} + K_{21}^{(2)} + K_{43}^{(3)}\right)d_1 + \left(K_{44}^{(1)} + K_{22}^{(2)} + K_{44}^{(3)}\right)d_2$$

These equations can be written compactly as:

$$\left\{\begin{array}{c} P_1 \\ P_2 \end{array}\right\} = \underbrace{\begin{bmatrix} K_{33}^{(1)} + K_{11}^{(2)} + K_{33}^{(3)} & K_{34}^{(1)} + K_{12}^{(2)} + K_{34}^{(3)} \\ K_{43}^{(1)} + K_{21}^{(2)} + K_{43}^{(3)} & K_{44}^{(1)} + K_{22}^{(2)} + K_{44}^{(3)} \end{bmatrix}}_{\boldsymbol{K}_{\text{ff}}} \left\{\begin{array}{c} d_1 \\ d_2 \end{array}\right\}$$

$$\boxed{\boldsymbol{P} = \boldsymbol{K}_{\text{ff}}\,\boldsymbol{d}}$$

Now we can solve for unknown displacements in the 2 free DOFs

Why notation K_{ff}? Later...to match partitioning node

# Physical Interpretation of the Structure Stiffness Matrix

- The **structure stiffness matrix** K_{ff} has the same physical meaning as an element stiffness matrix, but at the **structure (joint) level**
- A stiffness coefficient $K_{ff,ij}$ represents:
  - The **joint force** at DOF i
  - Required to cause a **unit displacement** at DOF j
  - While **all other joint displacements are zero**
- Equivalently:
  - Each **column** of K_{ff} corresponds to a unit displacement pattern
  - The column entries are the **resulting joint forces** needed to enforce that displacement

Unit Displacement Method - Derive First Column of $\boldsymbol{K}_{ff}$ - Unit Displacement Method

- To obtain the **first column** of $\boldsymbol{K}_{ff}$:
  - Impose a **unit displacement** at the first free DOF:

$$d_1 = 1, \qquad d_2 = 0$$

  - All other joint displacements are held fixed
- The resulting joint forces (DOF 1, 2) define the **first column** of $\boldsymbol{K}_{ff}$
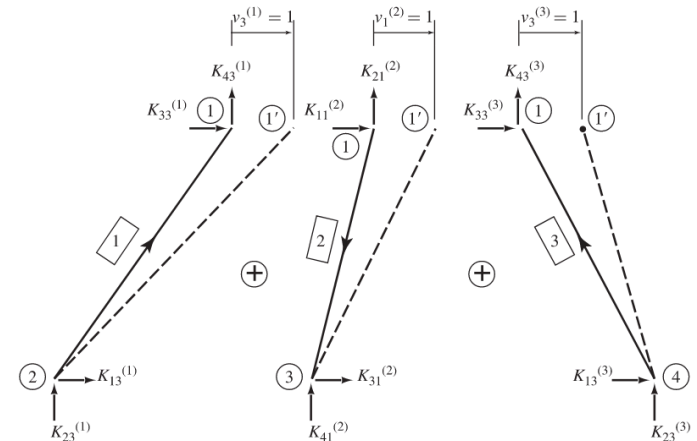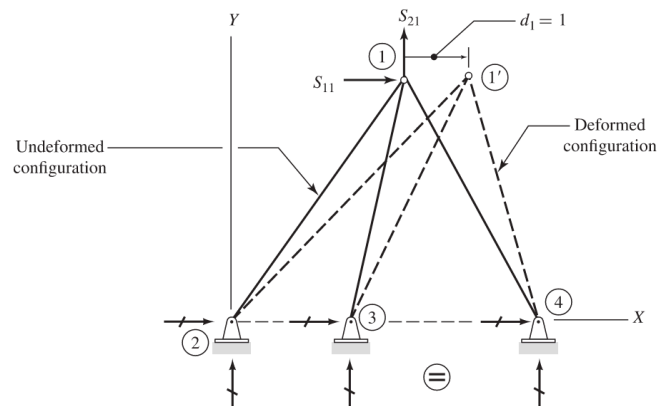
red = in direction of displacement

$$\begin{bmatrix} \boxed{K_{33}^{(1)} + K_{11}^{(2)} + K_{33}^{(3)}} & K_{34}^{(1)} + K_{12}^{(2)} + K_{34}^{(3)} \\[2mm] \boxed{K_{43}^{(1)} + K_{21}^{(2)} + K_{43}^{(3)}} & K_{44}^{(1)} + K_{22}^{(2)} + K_{44}^{(3)} \end{bmatrix}$$

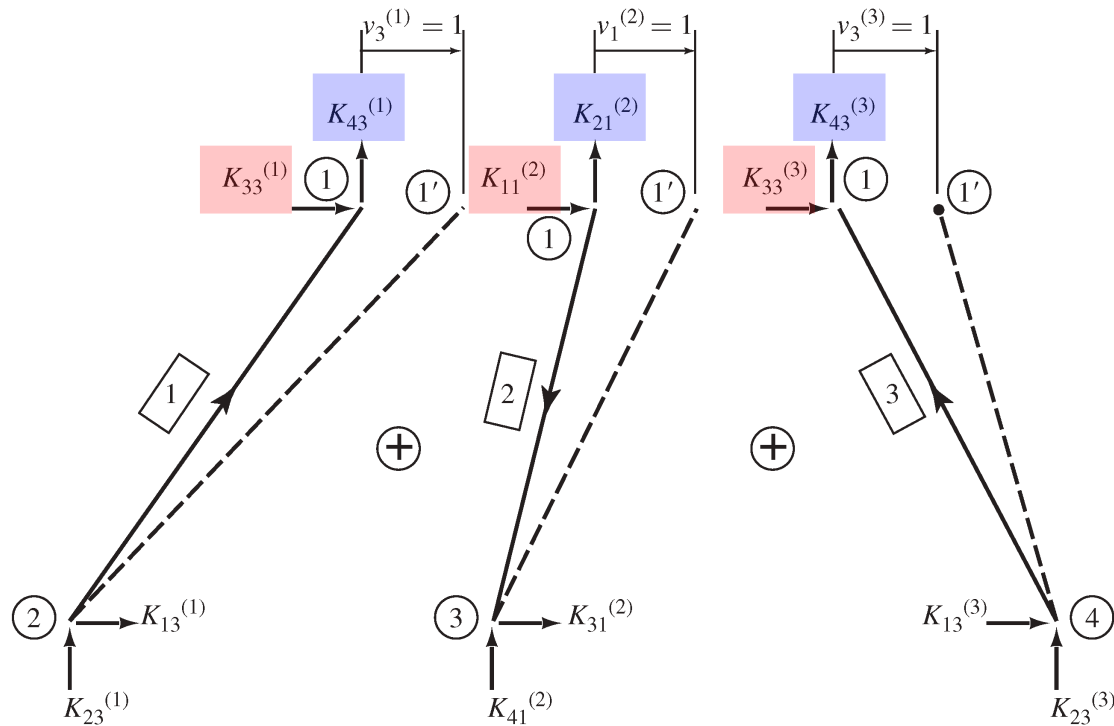# Unit Displacement Method - Derive Second Column of K_{ff}

- To obtain the **second column** of K_{ff}:
  - Impose a **unit displacement** at the first free DOF:

$$d_1 = 0, \qquad d_2 = 1$$

  - All other joint displacements are held fixed
- The resulting joint forces (DOF 1, 2) define the **second column** of K_{ff}
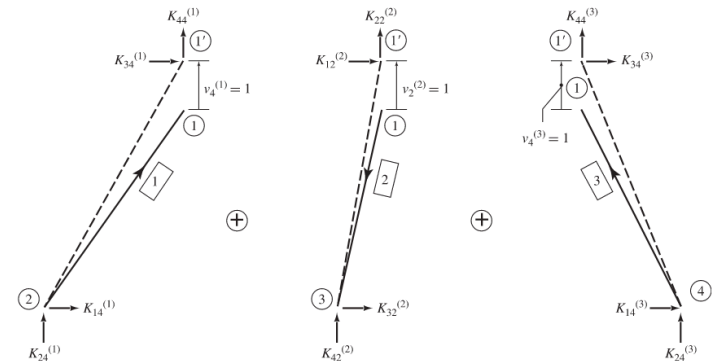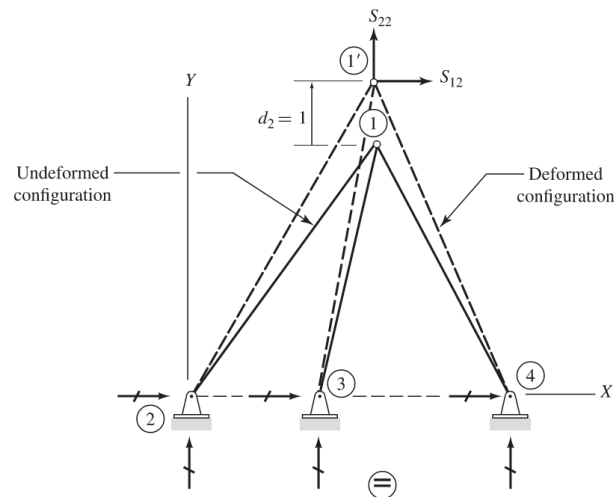
red = in direction of displacement
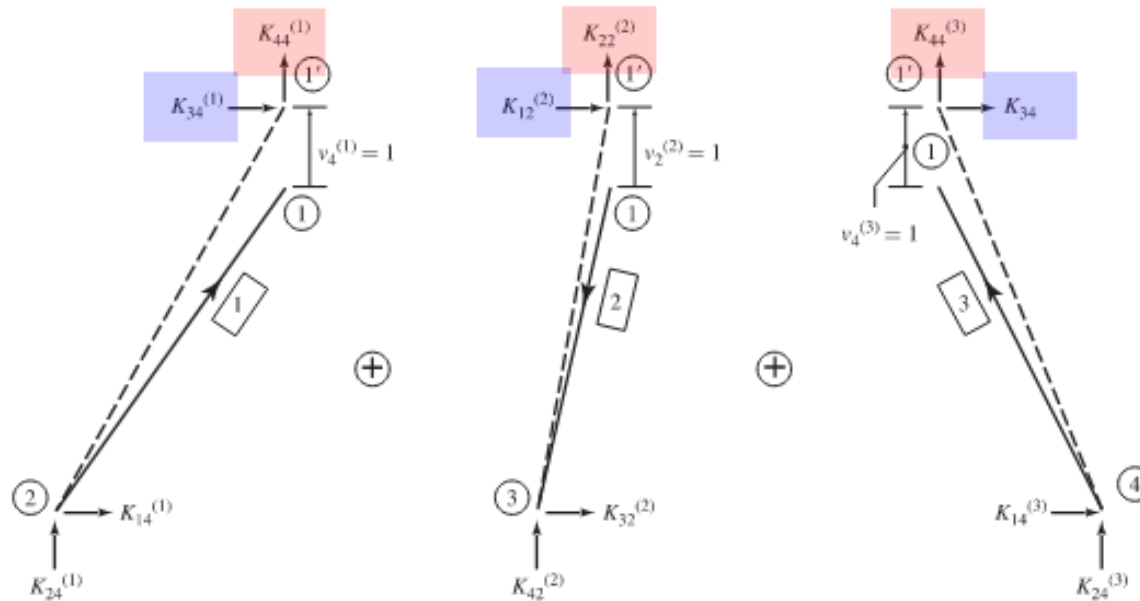
$$\begin{bmatrix} K_{33}^{(1)} + K_{11}^{(2)} + K_{33}^{(3)} & \boxed{K_{34}^{(1)} + K_{12}^{(2)} + K_{34}^{(3)}} \\ K_{43}^{(1)} + K_{21}^{(2)} + K_{43}^{(3)} & \boxed{K_{44}^{(1)} + K_{22}^{(2)} + K_{44}^{(3)}} \end{bmatrix}$$

Unit Displacement Method for More Complex Structures

Structure with 2 Free Nodes

1st Column

3rd Column

2nd Column

4th Column
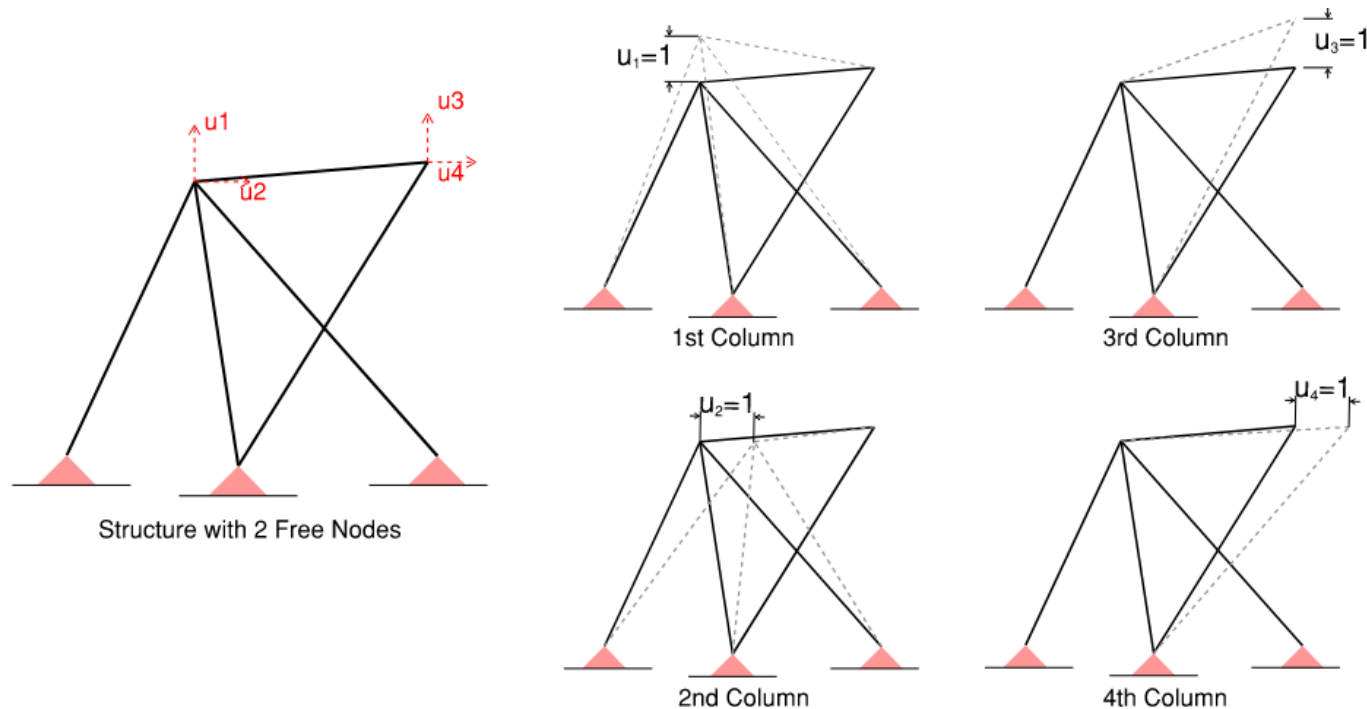
Force equilibrium and compatibility relationships are even harder to set up!

Part 4 — Assembling the Global Structure Stiffness Matrix

Direct Stiffness Method, Scatter Add

Why Not the Manual (Equilibrium + Compatibility) Approach?

- Becomes **tedious** very quickly
- Requires manually tracking:
  - Free vs constrained DOFs
  - Compatibility relationships
- Hard to implement **programmatically**
- Does **not scale** to larger or changing structures

Key idea: We want a method that is systematic, scalable, and algorithmic.

Reminder — Global Degrees of Freedom
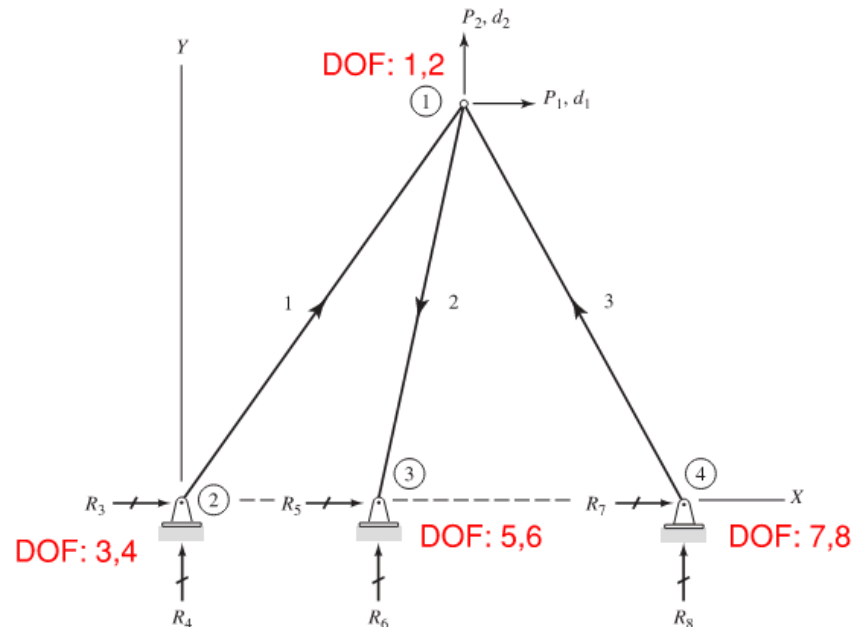
- Each node contributes two DOFs: (X, Y)
- 4 nodes → **8 global DOFs**
- Consistent numbering is essential for assembly

Map Element DOFs to Global DOFs

$$\mathbf{K}_1 = \begin{bmatrix} K_{11}^{(1)} & K_{12}^{(1)} & K_{13}^{(1)} & K_{14}^{(1)} \\ K_{21}^{(1)} & K_{22}^{(1)} & K_{23}^{(1)} & K_{24}^{(1)} \\ K_{31}^{(1)} & K_{32}^{(1)} & K_{33}^{(1)} & K_{34}^{(1)} \\ K_{41}^{(1)} & K_{42}^{(1)} & K_{43}^{(1)} & K_{44}^{(1)} \end{bmatrix} \begin{matrix} 3 \\ 4 \\ 1 \\ 2 \end{matrix}$$

(columns: 3, 4, 1, 2)

$$\mathbf{K}_2 = \begin{bmatrix} K_{11}^{(2)} & K_{12}^{(2)} & K_{13}^{(2)} & K_{14}^{(2)} \\ K_{21}^{(2)} & K_{22}^{(2)} & K_{23}^{(2)} & K_{24}^{(2)} \\ K_{31}^{(2)} & K_{32}^{(2)} & K_{33}^{(2)} & K_{34}^{(2)} \\ K_{41}^{(2)} & K_{42}^{(2)} & K_{43}^{(2)} & K_{44}^{(2)} \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 5 \\ 6 \end{matrix}$$

(columns: 1, 2, 5, 6)

$$\mathbf{K}_3 = \begin{bmatrix} K_{11}^{(3)} & K_{12}^{(3)} & K_{13}^{(3)} & K_{14}^{(3)} \\ K_{21}^{(3)} & K_{22}^{(3)} & K_{23}^{(3)} & K_{24}^{(3)} \\ K_{31}^{(3)} & K_{32}^{(3)} & K_{33}^{(3)} & K_{34}^{(3)} \\ K_{41}^{(3)} & K_{42}^{(3)} & K_{43}^{(3)} & K_{44}^{(3)} \end{bmatrix} \begin{matrix} 7 \\ 8 \\ 1 \\ 2 \end{matrix}$$

(columns: 7, 8, 1, 2)

Each element has **4 local DOFs** $(1, 2, 3, 4)$, which must be mapped to the correct **global DOFs** before assembly.

For this structure:

- **Element 1:** local $(1, 2, 3, 4)$ $\rightarrow$ global $(3, 4, 1, 2)$
- **Element 2:** local $(1, 2, 3, 4)$ $\rightarrow$ global $(1, 2, 5, 6)$
- **Element 3:** local $(1, 2, 3, 4)$ $\rightarrow$ global $(7, 8, 1, 2)$

This mapping determines **which rows and columns** of the global stiffness matrix each element stiffness contributes to.

Method 1: Assemble a Structure Matrix for the Free DOFs

In this example, only **global DOFs 1 and 2** are free; all other DOFs are constrained.

- Contributions from **each element** that map to DOFs 1 and 2 are **summed**
- This produces a reduced **structure stiffness matrix** for the free DOFs
- The result matches the stiffness matrix obtained earlier using the **manual equilibrium and compatibility** approach

For example, the $(1, 1)$ entry of the structure stiffness matrix is (matches manual):

$$K_{ff,11} = K_{33}^{(1)} + K_{11}^{(2)} + K_{33}^{(3)}$$

Method 2: Assemble the Full Global Stiffness Matrix

When assembling the **global structure stiffness matrix**, do **not** worry yet about which DOFs are constrained or free.

- **Assemble all DOFs first** into a single global matrix
- **Apply boundary conditions later** (supports, prescribed displacements)
- This separation is what makes the method **general, systematic, and scalable**

For the structure shown, there are **8 global DOFs**, so 8×8 stiffness matrix:

$$
\boldsymbol{K} =
\begin{bmatrix}
K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} & K_{17} & K_{18} \\
K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} & K_{27} & K_{28} \\
K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & K_{36} & K_{37} & K_{38} \\
K_{41} & K_{42} & K_{43} & K_{44} & K_{45} & K_{46} & K_{47} & K_{48} \\
K_{51} & K_{52} & K_{53} & K_{54} & K_{55} & K_{56} & K_{57} & K_{58} \\
K_{61} & K_{62} & K_{63} & K_{64} & K_{65} & K_{66} & K_{67} & K_{68} \\
K_{71} & K_{72} & K_{73} & K_{74} & K_{75} & K_{76} & K_{77} & K_{78} \\
K_{81} & K_{82} & K_{83} & K_{84} & K_{85} & K_{86} & K_{87} & K_{88}
\end{bmatrix}
$$

Each entry K_{ij} represents the **combined stiffness contribution** from all elements that connect global DOF j to global DOF i.

Element 1 — Contribution to Global Stiffness Matrix

Local → Global DOF mapping (Element 1):

$$(1, 2, 3, 4) \ \rightarrow \ (3, 4, 1, 2)$$

$$
\mathbf{K}_1 =
\begin{array}{cccc}
 & 3 & 4 & 1 & 2 \\
\begin{bmatrix}
K_{11}^{(1)} & K_{12}^{(1)} & K_{13}^{(1)} & K_{14}^{(1)} \\
K_{21}^{(1)} & K_{22}^{(1)} & K_{23}^{(1)} & K_{24}^{(1)} \\
K_{31}^{(1)} & K_{32}^{(1)} & K_{33}^{(1)} & K_{34}^{(1)} \\
K_{41}^{(1)} & K_{42}^{(1)} & K_{43}^{(1)} & K_{44}^{(1)}
\end{bmatrix}
&
\begin{matrix}
3 \\ 4 \\ 1 \\ 2
\end{matrix}
\end{array}
$$

$$
\mathbf{K} =
\begin{bmatrix}
K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} & K_{17} & K_{18} \\
K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} & K_{27} & K_{28} \\
K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & K_{36} & K_{37} & K_{38} \\
K_{41} & K_{42} & K_{43} & K_{44} & K_{45} & K_{46} & K_{47} & K_{48} \\
K_{51} & K_{52} & K_{53} & K_{54} & K_{55} & K_{56} & K_{57} & K_{58} \\
K_{61} & K_{62} & K_{63} & K_{64} & K_{65} & K_{66} & K_{67} & K_{68} \\
K_{71} & K_{72} & K_{73} & K_{74} & K_{75} & K_{76} & K_{77} & K_{78} \\
K_{81} & K_{82} & K_{83} & K_{84} & K_{85} & K_{86} & K_{87} & K_{88}
\end{bmatrix}
$$

Element 2 — Contribution to Global Stiffness Matrix

Local → Global DOF mapping (Element 2):

$$(1, 2, 3, 4) \ \rightarrow \ (1, 2, 5, 6)$$

$$
\mathbf{K}_2 =
\begin{bmatrix}
K_{11}^{(2)} & K_{12}^{(2)} & K_{13}^{(2)} & K_{14}^{(2)} \\
K_{21}^{(2)} & K_{22}^{(2)} & K_{23}^{(2)} & K_{24}^{(2)} \\
K_{31}^{(2)} & K_{32}^{(2)} & K_{33}^{(2)} & K_{34}^{(2)} \\
K_{41}^{(2)} & K_{42}^{(2)} & K_{43}^{(2)} & K_{44}^{(2)}
\end{bmatrix}
\begin{matrix} 1 \\ 2 \\ 5 \\ 6 \end{matrix}
$$

$$
\boldsymbol{K} =
\begin{bmatrix}
K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} & K_{17} & K_{18} \\
K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} & K_{27} & K_{28} \\
K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & K_{36} & K_{37} & K_{38} \\
K_{41} & K_{42} & K_{43} & K_{44} & K_{45} & K_{46} & K_{47} & K_{48} \\
K_{51} & K_{52} & K_{53} & K_{54} & K_{55} & K_{56} & K_{57} & K_{58} \\
K_{61} & K_{62} & K_{63} & K_{64} & K_{65} & K_{66} & K_{67} & K_{68} \\
K_{71} & K_{72} & K_{73} & K_{74} & K_{75} & K_{76} & K_{77} & K_{78} \\
K_{81} & K_{82} & K_{83} & K_{84} & K_{85} & K_{86} & K_{87} & K_{88}
\end{bmatrix}
$$

Element 3 — Contribution to Global Stiffness Matrix

Local → Global DOF mapping (Element 3):

$$(1, 2, 3, 4) \; \rightarrow \; (7, 8, 1, 2)$$

$$\mathbf{K}_3 = \begin{bmatrix} K_{11}^{(3)} & K_{12}^{(3)} & K_{13}^{(3)} & K_{14}^{(3)} \\ K_{21}^{(3)} & K_{22}^{(3)} & K_{23}^{(3)} & K_{24}^{(3)} \\ K_{31}^{(3)} & K_{32}^{(3)} & K_{33}^{(3)} & K_{34}^{(3)} \\ K_{41}^{(3)} & K_{42}^{(3)} & K_{43}^{(3)} & K_{44}^{(3)} \end{bmatrix} \begin{matrix} 7 \\ 8 \\ 1 \\ 2 \end{matrix}$$

$$\mathbf{K} = \begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} & K_{17} & K_{18} \\ K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} & K_{27} & K_{28} \\ K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & K_{36} & K_{37} & K_{38} \\ K_{41} & K_{42} & K_{43} & K_{44} & K_{45} & K_{46} & K_{47} & K_{48} \\ K_{51} & K_{52} & K_{53} & K_{54} & K_{55} & K_{56} & K_{57} & K_{58} \\ K_{61} & K_{62} & K_{63} & K_{64} & K_{65} & K_{66} & K_{67} & K_{68} \\ K_{71} & K_{72} & K_{73} & K_{74} & K_{75} & K_{76} & K_{77} & K_{78} \\ K_{81} & K_{82} & K_{83} & K_{84} & K_{85} & K_{86} & K_{87} & K_{88} \end{bmatrix}$$

Final Global Structure Stiffness Matrix

- Each K_{ij} may include **multiple element contributions**
- The matrix is **symmetric**
- Diagonal terms, K_{ii}, must be non-zero and positive for stability

$$
K = \begin{bmatrix}
K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} & K_{17} & K_{18} \\
K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} & K_{27} & K_{28} \\
K_{31} & K_{32} & K_{33} & K_{34} & 0 & 0 & 0 & 0 \\
K_{41} & K_{42} & K_{43} & K_{44} & 0 & 0 & 0 & 0 \\
K_{51} & K_{52} & 0 & 0 & K_{55} & K_{56} & 0 & 0 \\
K_{61} & K_{62} & 0 & 0 & K_{65} & K_{66} & 0 & 0 \\
K_{71} & K_{72} & 0 & 0 & 0 & 0 & K_{77} & K_{78} \\
K_{81} & K_{82} & 0 & 0 & 0 & 0 & K_{87} & K_{88}
\end{bmatrix}
$$

Interpreting Zeros in the Stiffness Matrix
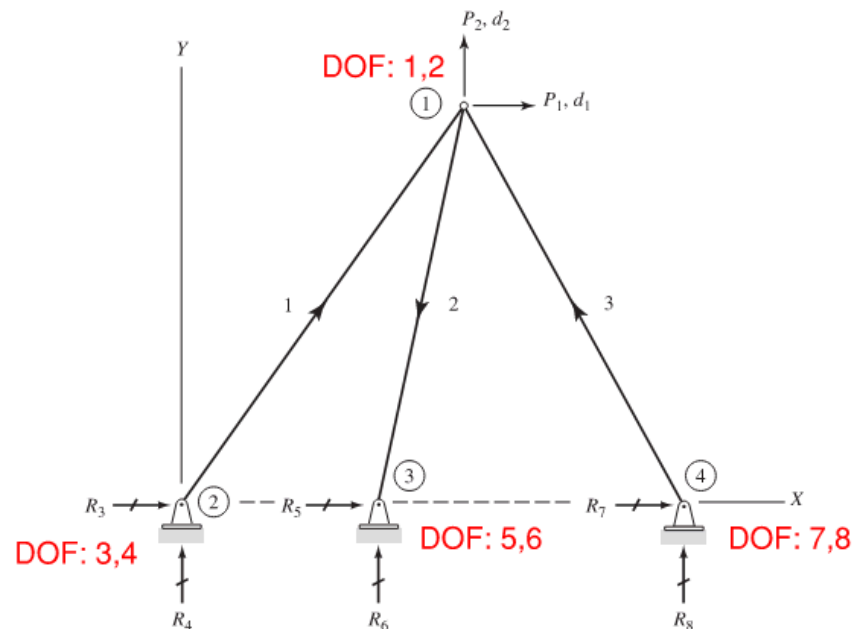
- Zero entry → no stiffness coupling
- Example: $K_{37} = 0$
- DOF 3 and DOF 7 share no connecting element

Part 5 — Constraints and Supports

Inverting the Global Stiffness Matrix

You might think: *we have* **K**, *so we can just solve*

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

and we're done.

However, for a **free (unconstrained) structure**, this step is **not yet valid**.

At this stage—before applying supports or boundary conditions—the global stiffness matrix typically contains **rigid-body modes** (motions that produce no strain energy). As a result:

- $\boldsymbol{K}$ is **singular** (not invertible)
- the structure still has **rigid-body motion** because we haven't applied supports / boundary conditions yet

Important note (about our 8 DOF example):

In our 8×8 formulation, the ability to invert **K** is an **artifact of how the matrix is set up**. The assembly already embeds enough *support-like stiffness*—intentionally or unintentionally—to suppress rigid-body motion.

As a result, the structure represented by this **K** is **not truly free**, even though we have not yet explicitly applied boundary conditions.

In general, however, a global stiffness matrix assembled for an unconstrained structure **will be singular**.
This is why the proper and robust approach is always to apply supports explicitly (via partitioning) before solving.

In [4]:

```python
import numpy as np

# Construct a singular 6x6 matrix
# Last row = sum of first two rows → linear dependence
K = np.array([
    [ 2, -1,  0,  0,  0,  0],
    [-1,  2, -1,  0,  0,  0],
    [ 0, -1,  2, -1,  0,  0],
    [ 0,  0, -1,  2, -1,  0],
    [ 0,  0,  0, -1,  2, -1],
    [ 1,  1, -1,  0,  0,  0]  # dependent row
], dtype=float)

try:
    np.linalg.inv(K)
except np.linalg.LinAlgError as e:
    print("Inversion failed:", e)
```

```
Inversion failed: Singular matrix
```

Why do we get a singular stiffness matrix?

Mathematically, an unconstrained structure admits non-zero displacement vectors $\mathbf{u} \neq \mathbf{0}$ such that

$$\mathbf{Ku} = \mathbf{0}$$

These displacement patterns are **rigid-body (zero-energy) modes**, meaning $\mathbf{K}$ has a **non-trivial nullspace** and is therefore **singular**.

What does this mean in practice?

As a result, the structure can move without inducing member deformation:

- **Rigid translation in** X
- **Rigid translation in** Y
- **Rigid-body rotation** (geometry-dependent)

These motions are **rigid-body modes**, not structural deformations.

Because rigid-body modes produce zero internal force, $\mathbf{K}$ cannot uniquely map applied loads to displacements. **Supports (boundary conditions)** remove these modes and render $\mathbf{K}$ invertible.

Boundary conditions: how supports enter the model

Supports and constraints are enforced by specifying **known displacements** (a.k.a. prescribed DOFs).

Common idealized supports (2D truss):

- **Pin support**: Prescribes both translations: $u_{ix} = 0$ and $u_{iy} = 0$
- **Roller support**: Prescribes one translation and allows the other:

Prescribed displacement (settlement / actuation):

- You can prescribe a non-zero value, e.g., $u_{iy} = -2 \text{ mm}$ at a support.

Why this fixes singularity:

- Constraints remove rigid-body modes by preventing global translations/rotations.
- The remaining free DOFs correspond to true structural deformation.

What we do next (workflow)

We do **not** modify the element --> global matrix formulation to handle supports.

Instead, the Direct Stiffness Method proceeds like this:

1. Assemble the **full** global matrix K and global load vector F
2. Identify which DOFs are **free** vs **restrained**
3. Enforce boundary conditions by **partitioning** (or equivalently, by eliminating restrained DOFs)
4. Solve only for the unknown (free) displacements
5. Recover reactions at restrained DOFs

So supports are handled cleanly at the **system equation** level — via partitioning (next section).

Part 6 — Partitioning the Matrix

Free vs Restrained DOFs

Core idea — Partition DOFs (free vs restrained)

Separate the global degrees of freedom into two sets:

- **Free DOFs** f: unknown displacements (to be solved)
- **Restrained DOFs** r: prescribed displacements (from supports or constraints)

Reorder the global displacement and force vectors so that all free DOFs come first:

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_f \\ \boldsymbol{u}_r \end{bmatrix}, \qquad \boldsymbol{F} = \begin{bmatrix} \boldsymbol{F}_f \\ \boldsymbol{F}_r \end{bmatrix}$$

Apply the **same reordering** to the global stiffness matrix to obtain a partitioned system:

$$\begin{bmatrix} \boldsymbol{K}_{ff} & \boldsymbol{K}_{fr} \\ \boldsymbol{K}_{rf} & \boldsymbol{K}_{rr} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_f \\ \boldsymbol{u}_r \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_f \\ \boldsymbol{F}_r \end{bmatrix}$$

Physical meaning K_{ff} etc..

Note: In our **8×8 truss example**, the system already appears partitioned because the two free global DOFs were labeled **1 and 2**, placing them in the correct positions from the start.

To illustrate the **general case**, the next slides use a separate **6×6 example** in which free and restrained DOFs are initially **intermixed**. The goal there is to show how the system is **reordered** so that all free DOFs appear first.

Step 0 — Original ordering (free DOFs = 3, 5)

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \boxed{u_3} \\ u_4 \\ \boxed{u_5} \\ u_6 \end{bmatrix}, \qquad \boldsymbol{F} = \begin{bmatrix} F_1 \\ F_2 \\ \boxed{F_3} \\ F_4 \\ \boxed{F_5} \\ F_6 \end{bmatrix}$$

The corresponding global stiffness matrix has rows and columns associated with the same DOFs. Rows and columns for free DOFs (3 and 5) are highlighted.

$$\boldsymbol{K} = \begin{bmatrix} K_{11} & K_{12} & \boxed{K_{13}} & K_{14} & \boxed{K_{15}} & K_{16} \\ K_{21} & K_{22} & \boxed{K_{23}} & K_{24} & \boxed{K_{25}} & K_{26} \\ \boxed{K_{31}} & \boxed{K_{32}} & \boxed{K_{33}} & \boxed{K_{34}} & \boxed{K_{35}} & \boxed{K_{36}} \\ K_{41} & K_{42} & \boxed{K_{43}} & K_{44} & \boxed{K_{45}} & K_{46} \\ \boxed{K_{51}} & \boxed{K_{52}} & \boxed{K_{53}} & \boxed{K_{54}} & \boxed{K_{55}} & \boxed{K_{56}} \\ K_{61} & K_{62} & \boxed{K_{63}} & K_{64} & \boxed{K_{65}} & K_{66} \end{bmatrix}$$

Step 1 — Bring DOF 3 to the top (swap DOFs 1 and 3)

Displacement and force vectors

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \boxed{u_3} \\ u_4 \\ \boxed{u_5} \\ u_6 \end{bmatrix} \longrightarrow \begin{bmatrix} \boxed{u_3} \\ u_2 \\ u_1 \\ u_4 \\ \boxed{u_5} \\ u_6 \end{bmatrix}, \qquad \boldsymbol{F} = \begin{bmatrix} F_1 \\ F_2 \\ \boxed{F_3} \\ F_4 \\ \boxed{F_5} \\ F_6 \end{bmatrix} \longrightarrow \begin{bmatrix} \boxed{F_3} \\ F_2 \\ F_1 \\ F_4 \\ \boxed{F_5} \\ F_6 \end{bmatrix}$$

Stiffness matrix (rows 1 ↔ 3, columns 1 ↔ 3)

$$\boldsymbol{K}^{(1)} = \begin{bmatrix} \boxed{K_{33}} & \boxed{K_{32}} & \boxed{K_{31}} & \boxed{K_{34}} & \boxed{K_{35}} & \boxed{K_{36}} \\ \boxed{K_{23}} & K_{22} & K_{21} & K_{24} & \boxed{K_{25}} & K_{26} \\ \boxed{K_{13}} & K_{12} & K_{11} & K_{14} & \boxed{K_{15}} & K_{16} \\ \boxed{K_{43}} & K_{42} & K_{41} & K_{44} & \boxed{K_{45}} & K_{46} \\ \boxed{K_{53}} & \boxed{K_{52}} & \boxed{K_{51}} & \boxed{K_{54}} & \boxed{K_{55}} & \boxed{K_{56}} \\ \boxed{K_{63}} & K_{62} & K_{61} & K_{64} & \boxed{K_{65}} & K_{66} \end{bmatrix}$$

Step 2 — Bring DOF 5 to position 2 (swap DOFs 2 and 5)

Displacement and force vectors

$$\boldsymbol{u} = \begin{bmatrix} \boxed{u_3} \\ u_2 \\ u_1 \\ u_4 \\ \boxed{u_5} \\ u_6 \end{bmatrix} \longrightarrow \begin{bmatrix} \boxed{u_3} \\ \boxed{u_5} \\ u_1 \\ u_4 \\ u_2 \\ u_6 \end{bmatrix}, \qquad \boldsymbol{F} = \begin{bmatrix} \boxed{F_3} \\ F_2 \\ F_1 \\ F_4 \\ \boxed{F_5} \\ F_6 \end{bmatrix} \longrightarrow \begin{bmatrix} \boxed{F_3} \\ \boxed{F_5} \\ F_1 \\ F_4 \\ F_2 \\ F_6 \end{bmatrix}$$

Stiffness matrix (rows 2 ↔ 5, columns 2 ↔ 5)

$$\boldsymbol{K}^{(2)} = \begin{bmatrix} \boxed{K_{33}} & \boxed{K_{35}} & \boxed{K_{31}} & \boxed{K_{34}} & \boxed{K_{32}} & \boxed{K_{36}} \\ \boxed{K_{53}} & \boxed{K_{55}} & \boxed{K_{51}} & \boxed{K_{54}} & \boxed{K_{52}} & \boxed{K_{56}} \\ \boxed{K_{13}} & \boxed{K_{15}} & K_{11} & K_{14} & K_{12} & K_{16} \\ \boxed{K_{43}} & \boxed{K_{45}} & K_{41} & K_{44} & K_{42} & K_{46} \\ \boxed{K_{23}} & \boxed{K_{25}} & K_{21} & K_{24} & K_{22} & K_{26} \\ \boxed{K_{63}} & \boxed{K_{65}} & K_{61} & K_{64} & K_{62} & K_{66} \end{bmatrix}$$

Partitioned form (read directly)

After reordering, the free DOFs come first. For our example, the free set is $f = \{3, 5\}$ and the restrained set is $r = \{1, 4, 2, 6\}$, so

$$
\boldsymbol{u} = \left[\begin{array}{c} \boldsymbol{u}_f \\ \hdashline \boldsymbol{u}_r \end{array}\right] = \begin{bmatrix} u_3 \\ u_5 \\ \hdashline u_1 \\ u_4 \\ u_2 \\ u_6 \end{bmatrix}, \qquad
\boldsymbol{F} = \left[\begin{array}{c} \boldsymbol{F}_f \\ \hdashline \boldsymbol{F}_r \end{array}\right] = \begin{bmatrix} F_3 \\ F_5 \\ \hdashline F_1 \\ F_4 \\ F_2 \\ F_6 \end{bmatrix}
$$

$$
\boldsymbol{K} = \left[\begin{array}{c:c} \boldsymbol{K}_{ff} & \boldsymbol{K}_{fr} \\ \hdashline \boldsymbol{K}_{rf} & \boldsymbol{K}_{rr} \end{array}\right] = \left[\begin{array}{cc:cccc}
K_{33} & K_{35} & K_{31} & K_{34} & K_{32} & K_{36} \\
K_{53} & K_{55} & K_{51} & K_{54} & K_{52} & K_{56} \\ \hdashline
K_{13} & K_{15} & K_{11} & K_{14} & K_{12} & K_{16} \\
K_{43} & K_{45} & K_{41} & K_{44} & K_{42} & K_{46} \\
K_{23} & K_{25} & K_{21} & K_{24} & K_{22} & K_{26} \\
K_{63} & K_{65} & K_{61} & K_{64} & K_{62} & K_{66}
\end{array}\right]
$$

Here the dashed line shows the split between free and restrained DOFs:

- top-left block: $\boldsymbol{K}_{ff}$ (free–free)
- top-right block: $\boldsymbol{K}_{fr}$ (free–restrained)
- bottom-left block: $\boldsymbol{K}_{rf}$ (restrained–free)
- bottom-right block: $\boldsymbol{K}_{rr}$ (restrained–restrained)

Part 7 — Solving Global Displacements and Forces

free node displacements and restraint forces

Expand the Partitioned System

Starting from the block system

$$\begin{bmatrix} \boldsymbol{K}_{ff} & \boldsymbol{K}_{fr} \\ \boldsymbol{K}_{rf} & \boldsymbol{K}_{rr} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_f \\ \boldsymbol{u}_r \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_f \\ \boldsymbol{F}_r \end{bmatrix}$$

expanding the matrix–vector multiplication gives two coupled equations:

$$\boldsymbol{K}_{ff}\boldsymbol{u}_f + \boldsymbol{K}_{fr}\boldsymbol{u}_r = \boldsymbol{F}_f \tag{1}$$

$$\boldsymbol{K}_{rf}\boldsymbol{u}_f + \boldsymbol{K}_{rr}\boldsymbol{u}_r = \boldsymbol{F}_r \tag{2}$$

The first equation governs equilibrium at the **free DOFs**,
and the second governs equilibrium at the **restrained DOFs**.

Solving Equation (1) for the Free DOFs

We are interested in solving for the **unknown displacements** at the free DOFs, $\boldsymbol{u}_f$.

Starting from Equation (1): $\boldsymbol{K}_{ff}\boldsymbol{u}_f + \boldsymbol{K}_{fr}\boldsymbol{u}_r = \boldsymbol{F}_f$

Rearrange to isolate the unknowns:

$$\boldsymbol{K}_{ff}\boldsymbol{u}_f = \boldsymbol{F}_f - \boldsymbol{K}_{fr}\boldsymbol{u}_r$$

Provided that $\boldsymbol{K}_{ff}$ is invertible, the solution is:

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

Typically, the restrained displacements $\boldsymbol{u}_r$ are **known** from boundary conditions (often $\boldsymbol{u}_r = \boldsymbol{0}$). In that common case, the expression simplifies to:

$$\boxed{\boldsymbol{u}_f = \boldsymbol{K}_{ff}^{-1}\boldsymbol{F}_f}$$

Solve in practice to obtain the global displacements at the free DOFs.

Solving Equation (2) for the Restrained Forces

Once the free displacements $\boldsymbol{u}_f$ have been computed, we can determine the **forces at the restrained DOFs** (support reactions).

Starting from Equation (2): $\boldsymbol{K}_{rf}\boldsymbol{u}_f + \boldsymbol{K}_{rr}\boldsymbol{u}_r = \boldsymbol{F}_r$

At the restrained DOFs, the displacements $\boldsymbol{u}_r$ are **known** from the boundary conditions (often $\boldsymbol{u}_r = \boldsymbol{0}$). Substituting these known values gives a direct expression for the reaction forces:

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

In the common case where the supports are fixed and $\boldsymbol{u}_r = \boldsymbol{0}$, this simplifies to:

$$\boxed{\boldsymbol{F}_r = \boldsymbol{K}_{rf}\boldsymbol{u}_f}$$

This step **does not require solving a linear system** (i.e., no inverse needed).
The reaction forces are obtained directly by evaluating this expression after $\boldsymbol{u}_f$ is known.

Summary — What We Have Solved

At this point, the global system has been fully resolved.

We have determined the two previously unknown quantities:

- **Free nodal displacements**, u_f
  (obtained by solving the reduced system)

- **Support reaction forces**, F_r
  (computed directly once u_f is known)

Together, these results provide the complete displacement and force state of the structure, satisfying equilibrium, compatibility, and the imposed boundary conditions.

Sanity checks after solving

1) Stability

- If the structure is properly constrained, $\boldsymbol{K}_{ff}$ should be non-singular
- If it is still singular, you likely have:
  - insufficient supports (rigid-body mode remains), or
  - a disconnected node / element connectivity error

2) Equilibrium

- Reconstruct the full displacement vector $\boldsymbol{u}$ by inserting $\boldsymbol{u}_f$ (just calculated) and $\boldsymbol{u}_r$
- Compute $\boldsymbol{K}\boldsymbol{u}$ and compare to $\boldsymbol{F}$
- Differences should be near numerical tolerance (floating point error)

3) Reactions

- Check reaction directions and magnitudes against intuition
- Sum of reactions should balance applied loads (global equilibrium)

Part 8 — Recover Element Forces

Back to Local

Up to this point, we have solved for the **global nodal displacements** of the structure. While this information is essential for equilibrium, it is **not** what we design members for.

Structural design is performed at the **element level**, using:

- axial forces
- stresses
- internal force resultants
  expressed in each element's **local coordinate system**.

This step bridges analysis and design.

Why Go Back to the Element Level?

- Global displacements describe how the **structure moves**
- Local element forces describe how **members carry load**
- Design checks (strength, buckling, serviceability) are based on **element forces**, not global DOFs

Therefore, we must recover **element-level displacements and forces** from the global solution.

Element Force Recovery — Overview

For each element, we perform the following steps:

1. **Extract element global displacements**
   Assemble the element displacement vector from the global solution.

2. **Transform to local coordinates**
   Convert global displacements to the element's local axis.

3. **Compute local end forces**
   Use the element stiffness matrix to compute internal forces.

4. **Evaluate axial force**
   Extract the axial force used directly in design.

Step 1 — Extract Element Global Displacements

From the global displacement vector $\boldsymbol{u}$, collect the DOFs associated with element e:

$$\boldsymbol{u_e} = \begin{bmatrix} u_{i,x} \\ u_{i,y} \\ u_{j,x} \\ u_{j,y} \end{bmatrix}$$

This vector contains the **global displacements** of the element's end nodes.

Step 2 — Transform to Local Coordinates

Transform the element displacement vector to the local coordinate system:

$$u'_e = T_e u_e$$

where T_e is the **element transformation matrix**, for that specific element defined by the element orientation.

This step aligns the displacements with the element's axial direction.

Step 3 — Compute Local End Forces

Use the local stiffness matrix to compute the internal end forces:

$$\boldsymbol{f}'_e = \boldsymbol{k}'_e \boldsymbol{u}'_e$$

These are the **local element forces** (axial forces at each end). For a truss element you should get the axial force in the element

Step 4 — Compute Axial Stress

Once the axial force is known, the **axial stress** follows as:

$$\sigma = \frac{N}{A}$$

This is what you design for:

- axial stress limits
- tension vs. compression
- buckling checks for compression members

Step 5 (Optional) — Member End Forces in Global Coordinates

This step is **not required for design**, but it can be useful for interpretation and post-processing.

So far, member forces have been computed in the **local coordinate system**, which is where axial behavior is defined.
In some cases, however, it is helpful to express the **member end forces in the global** (X, Y) **frame**, for example:

- to visualize force flow in the structure
- to check nodal equilibrium in global directions
- to compare with externally applied loads

The local member force vector can be rotated back to global coordinates as:

$$\boldsymbol{f}_e = \boldsymbol{T}_e^\mathsf{T} \boldsymbol{f}_e'$$

For a **perfectly horizontal member**, the local and global axes coincide, and this transformation leaves the forces unchanged.

Part 9 — DSM Summary

Forward Pass — Structural Analysis

1. **Defining Structure**

   - Node numbering and coordinates
   - Global DOF numbering
   - Element connectivity
   - Restraints and Applied Forces

2. **Element-Level Stiffness Matrix**

   - Compute geometry: L, c, s
   - Build transformation matrix, $\boldsymbol{T}$
   - Compute global element stiffness:

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

3. **Assemble global stiffness matrix and force vector**

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

Forward Pass — Structural Analysis, cont…

4. **Apply boundary conditions**

- Partition DOFs into free (f) and restrained (r)

5. **Solve for unknown displacements**

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

6. **Recover support reactions**

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

Backward Pass — Element Recovery and Design

7. **Extract element global displacement vectors**

- For each member, collect the relevant entries from $\boldsymbol{u}$ to form $\boldsymbol{u}_e$

8. **Transform displacements to local coordinates**

$$\boldsymbol{u}' = \boldsymbol{T}\,\boldsymbol{u}_e$$

9. **Compute local element end forces**

$$\boldsymbol{f}' = \boldsymbol{k}'\,\boldsymbol{u}'$$

10. **Compute axial force and stress (design quantities)**

$$\sigma = \frac{N}{A}$$

11. **(Optional) Express element forces in global coordinates**

$$\boldsymbol{f} = \boldsymbol{T}^{\mathsf{T}}\boldsymbol{f}'$$

Why DSM Is Powerful

- **Static indeterminacy is not an issue**
  - Any number of members can be handled
- **Equilibrium is enforced automatically**
  - Through the global stiffness equations
- **Compatibility is built in**
  - Via shared nodal displacements
- The method scales cleanly to:
  - large structures
  - complex geometries
  - numerical implementation

> *DSM replaces manual equilibrium and compatibility equations with a single, systematic matrix framework.*

Wrap-Up

Last lecture you built the DSM pipeline for trusses:

- local bar stiffness $\rightarrow$ transformation $\rightarrow$ global element stiffness

This Lecture you:

- assembly $\rightarrow$ constraints $\rightarrow$ solve $\rightarrow$ member force recovery

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