CIEM5000: Structural Engineering Base

Matrix Method — Final Details

Tom van Woudenberg, Iuri Rocha

The Matrix Method

Main steps:

- Extract element matrices
- Impose nodal equilibrium
- Impose boundary conditions
- Solve for unknown displacements
- Postprocess results

This week:

- Element loads
- Non-zero Dirichlet boundary conditions in two different ways
- Postprocessing: support reactions and element fields
- Matrix method versus FEM — parallels and differences
- Example: A fully-resolved example by hand
- Workshop: Wrap up the code and solve a frame structure

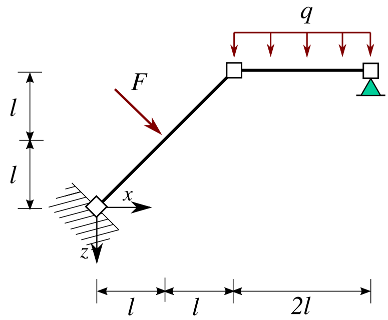
Element loads

The matrix method is a discrete approach

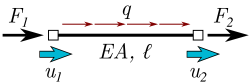
- Nodal loads treated easily
- What if we have loads applied inside elements?

A number of approaches to handle this:

- Further discretization (seldom helps)
- ODE approach
- Work-based approach



Element loads — ODE approach

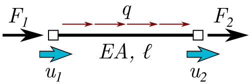


We can follow the same steps as before:

- General solution for the ODE:

$$EA\frac{\mathrm{d}^2u}{\mathrm{d}x^2} = -q \quad \Rightarrow \quad u(x) = -\frac{qx^2}{2EA} + C_1x + C_2$$

Element loads — ODE approach



We can follow the same steps as before:

- General solution for the ODE:

$$EA\frac{\mathrm{d}^2u}{\mathrm{d}x^2} = -q \quad \Rightarrow \quad u(x) = -\frac{qx^2}{2EA} + C_1x + C_2$$

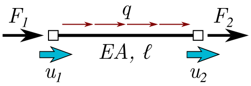
- Boundary conditions and final solution:

$$u(0) = u_1 \quad u(\ell) = u_2 \quad \Rightarrow \quad C_1 = \frac{q\ell}{2EA} + \frac{u_2 - u_1}{\ell} \quad C_2 = u_1$$

$$u(x) = \frac{q}{2EA}\left(\ell x - x^2\right) + u_1\left(1 - \frac{x}{\ell}\right) + \frac{u_2 x}{\ell}$$

TU Delft

Element loads — ODE approach



We can follow the same steps as before:

- General solution for the ODE:

$$EA\frac{\mathrm{d}^2u}{\mathrm{d}x^2} = -q \quad \Rightarrow \quad u(x) = -\frac{qx^2}{2EA} + C_1x + C_2$$

- Boundary conditions and final solution:

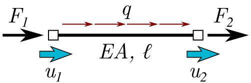
$$u(0) = u_1 \quad u(\ell) = u_2 \quad \Rightarrow \quad C_1 = \frac{q\ell}{2EA} + \frac{u_2 - u_1}{\ell} \quad C_2 = u_1$$

$$u(x) = \frac{q}{2EA}\left(\ell x - x^2\right) + u_1\left(1 - \frac{x}{\ell}\right) + \frac{u_2 x}{\ell}$$

- Relate forces at the edges with internal stresses:

$$N(x) = \frac{EA}{\ell}\left(u_2 - u_1\right) + \frac{q\ell - 2qx}{2} \qquad F_1 = -N_1 \quad F_2 = N_2$$

Element loads — ODE approach



We can follow the same steps as before:

- General solution for the ODE:

$$EA\frac{\mathrm{d}^2u}{\mathrm{d}x^2} = -q \quad \Rightarrow \quad u(x) = -\frac{qx^2}{2EA} + C_1x + C_2$$

- Boundary conditions and final solution:

$$u(0) = u_1 \quad u(\ell) = u_2 \quad \Rightarrow \quad C_1 = \frac{q\ell}{2EA} + \frac{u_2 - u_1}{\ell} \quad C_2 = u_1$$

$$u(x) = \frac{q}{2EA}\left(\ell x - x^2\right) + u_1\left(1 - \frac{x}{\ell}\right) + \frac{u_2x}{\ell}$$

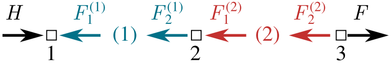
- Relate forces at the edges with internal stresses:

$$N(x) = \frac{EA}{\ell}\left(u_2 - u_1\right) + \frac{q\ell - 2qx}{2} \qquad F_1 = -N_1 \quad F_2 = N_2$$

- Relate forces and displacements, but now an extra term appears:

$$\frac{EA}{\ell}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix} - \begin{bmatrix}\frac{q\ell}{2}\\ \frac{q\ell}{2}\end{bmatrix} = \begin{bmatrix}F_1\\ F_2\end{bmatrix}$$

Dealing with equivalent loads



The new term is an equivalent nodal load:

- Element loads $\Rightarrow$ nodal loads
- Force equilibrium at the nodes therefore changes a bit:

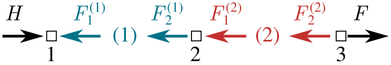
$$-\sum_e \mathbf{f}^e + \mathbf{f}_{\text{nodal}} = \mathbf{0}$$

$$-\sum_e \left(\mathbf{K}^e \mathbf{u}^e - \mathbf{f}_{\text{eq}}^e\right) + \mathbf{f}_{\text{nodal}} = \mathbf{0}$$

$$\sum_e \left(\mathbf{K}^e \mathbf{u}^e\right) = \mathbf{f}_{\text{nodal}} + \sum_e \mathbf{f}_{\text{eq}}^e$$

TUDelft

Dealing with equivalent loads



The new term is an equivalent nodal load:

- Element loads ⇒ nodal loads
- Force equilibrium at the nodes therefore changes a bit:

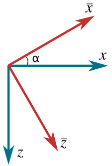
$$-\sum_e \mathbf{f}^e + \mathbf{f}_{\text{nodal}} = \mathbf{0}$$

$$-\sum_e \left(\mathbf{K}^e\mathbf{u}^e - \mathbf{f}^e_{\text{eq}}\right) + \mathbf{f}_{\text{nodal}} = \mathbf{0}$$

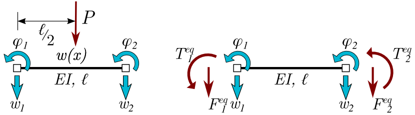
$$\sum_e \left(\mathbf{K}^e\mathbf{u}^e\right) = \mathbf{f}_{\text{nodal}} + \sum_e \mathbf{f}^e_{\text{eq}}$$

Remember, this is a force in the global coordinate system!

$$\mathbf{f}_{\text{eq}} = \mathbf{T}^{\mathrm{T}}\bar{\mathbf{f}}_{\text{eq}}$$



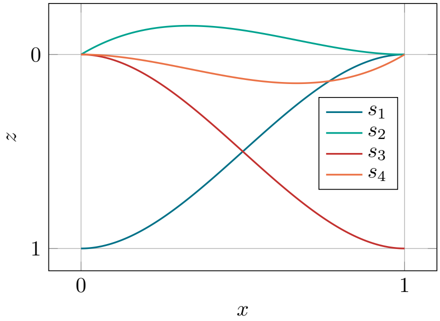
Work-based element loads



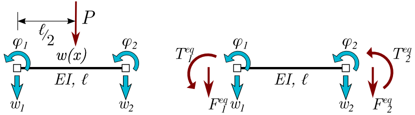
Euler-Bernoulli bending, point load at midspan:

- Displacement field for arbitrary DOFs (ODE with $q = 0$):

$$w(x) = \underbrace{\left(\frac{2x^3}{\ell^3} - \frac{3x^2}{\ell^2} + 1\right)}_{s_1} w_1 + \underbrace{\left(-\frac{x^3}{\ell^2} + \frac{2x^2}{\ell} - x\right)}_{s_2} \varphi_1 + \underbrace{\left(-\frac{2x^3}{\ell^3} + \frac{3x^2}{\ell^2}\right)}_{s_3} w_2 + \underbrace{\left(-\frac{x^3}{\ell^2} + \frac{x^2}{\ell}\right)}_{s_4} \varphi_2$$



Work-based element loads



Euler-Bernoulli bending, point load at midspan:

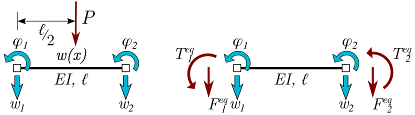
- Displacement field for arbitrary DOFs (ODE with $q = 0$):

$$w(x) = \underbrace{\left(\frac{2x^3}{\ell^3} - \frac{3x^2}{\ell^2} + 1\right)}_{s_1} w_1 + \underbrace{\left(-\frac{x^3}{\ell^2} + \frac{2x^2}{\ell} - x\right)}_{s_2} \varphi_1 + \underbrace{\left(-\frac{2x^3}{\ell^3} + \frac{3x^2}{\ell^2}\right)}_{s_3} w_2 + \underbrace{\left(-\frac{x^3}{\ell^2} + \frac{x^2}{\ell}\right)}_{s_4} \varphi_2$$

- Work performed by edge forces:

$$W_F = F_1^{\mathrm{eq}} w_1 + T_1^{\mathrm{eq}} \varphi_1 + F_2^{\mathrm{eq}} w_2 + T_2^{\mathrm{eq}} \varphi_2$$

Work-based element loads



Euler-Bernoulli bending, point load at midspan:

- Displacement field for arbitrary DOFs (ODE with $q = 0$):

$$w(x) = \underbrace{\left(\frac{2x^3}{\ell^3} - \frac{3x^2}{\ell^2} + 1\right)}_{s_1} w_1 + \underbrace{\left(-\frac{x^3}{\ell^2} + \frac{2x^2}{\ell} - x\right)}_{s_2} \varphi_1 + \underbrace{\left(-\frac{2x^3}{\ell^3} + \frac{3x^2}{\ell^2}\right)}_{s_3} w_2 + \underbrace{\left(-\frac{x^3}{\ell^2} + \frac{x^2}{\ell}\right)}_{s_4} \varphi_2$$

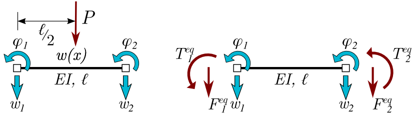
- Work performed by edge forces:

$$W_F = F_1^{\mathrm{eq}} w_1 + T_1^{\mathrm{eq}} \varphi_1 + F_2^{\mathrm{eq}} w_2 + T_2^{\mathrm{eq}} \varphi_2$$

- Work performed by P under the same displacement:

$$W_q = Pw(\ell/2) = \left(Ps_1(\ell/2)\right) w_1 + \left(Ps_2(\ell/2)\right) \varphi_1 + \left(Ps_3(\ell/2)\right) w_2 + \left(Ps_4(\ell/2)\right) \varphi_2$$

Work-based element loads



Euler-Bernoulli bending, point load at midspan:

- Displacement field for arbitrary DOFs (ODE with $q = 0$):

$$w(x) = \underbrace{\left(\frac{2x^3}{\ell^3} - \frac{3x^2}{\ell^2} + 1\right)}_{s_1} w_1 + \underbrace{\left(-\frac{x^3}{\ell^2} + \frac{2x^2}{\ell} - x\right)}_{s_2} \varphi_1 + \underbrace{\left(-\frac{2x^3}{\ell^3} + \frac{3x^2}{\ell^2}\right)}_{s_3} w_2 + \underbrace{\left(-\frac{x^3}{\ell^2} + \frac{x^2}{\ell}\right)}_{s_4} \varphi_2$$

- Work performed by edge forces:

$$W_F = F_1^{\mathrm{eq}} w_1 + T_1^{\mathrm{eq}} \varphi_1 + F_2^{\mathrm{eq}} w_2 + T_2^{\mathrm{eq}} \varphi_2$$

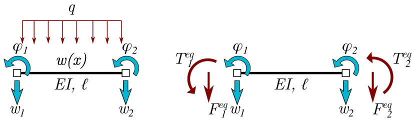
- Work performed by P under the same displacement:

$$W_q = P w(\ell/2) = \left(P s_1(\ell/2)\right) w_1 + \left(P s_2(\ell/2)\right) \varphi_1 + \left(P s_3(\ell/2)\right) w_2 + \left(P s_4(\ell/2)\right) \varphi_2$$

- Enforcing $W_F = W_q$ and isolating terms:

$$\mathbf{f}_{\mathrm{eq}} = \begin{bmatrix} F_1^{\mathrm{eq}} \\ T_1^{\mathrm{eq}} \\ F_2^{\mathrm{eq}} \\ T_2^{\mathrm{eq}} \end{bmatrix} = \begin{bmatrix} \frac{P}{2} \\ -\frac{P\ell}{8} \\ \frac{P}{2} \\ \frac{P\ell}{8} \end{bmatrix}$$

Work-based element loads

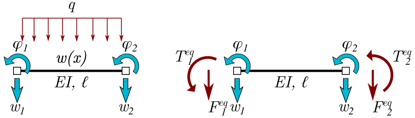


Euler-Bernoulli bending, constant distributed load:

- Displacement field for arbitrary DOFs (ODE with $q = 0$):

$$w(x) = \underbrace{\left(\frac{2x^3}{\ell^3} - \frac{3x^2}{\ell^2} + 1\right)}_{s_1} w_1 + \underbrace{\left(-\frac{x^3}{\ell^2} + \frac{2x^2}{\ell} - x\right)}_{s_2} \varphi_1 + \underbrace{\left(-\frac{2x^3}{\ell^3} + \frac{3x^2}{\ell^2}\right)}_{s_3} w_2 + \underbrace{\left(-\frac{x^3}{\ell^2} + \frac{x^2}{\ell}\right)}_{s_4} \varphi_2$$

Work-based element loads



Euler-Bernoulli bending, constant distributed load:

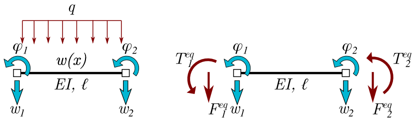
- Displacement field for arbitrary DOFs (ODE with $q = 0$):

$$w(x) = \underbrace{\left(\frac{2x^3}{\ell^3} - \frac{3x^2}{\ell^2} + 1\right)}_{s_1} w_1 + \underbrace{\left(-\frac{x^3}{\ell^2} + \frac{2x^2}{\ell} - x\right)}_{s_2} \varphi_1 + \underbrace{\left(-\frac{2x^3}{\ell^3} + \frac{3x^2}{\ell^2}\right)}_{s_3} w_2 + \underbrace{\left(-\frac{x^3}{\ell^2} + \frac{x^2}{\ell}\right)}_{s_4} \varphi_2$$

- Work performed by edge forces:

$$W_F = F_1^{\text{eq}} w_1 + T_1^{\text{eq}} \varphi_1 + F_2^{\text{eq}} w_2 + T_2^{\text{eq}} \varphi_2$$

Work-based element loads



Euler-Bernoulli bending, constant distributed load:

- Displacement field for arbitrary DOFs (ODE with $q = 0$):

$$w(x) = \underbrace{\left(\frac{2x^3}{\ell^3} - \frac{3x^2}{\ell^2} + 1\right)}_{s_1} w_1 + \underbrace{\left(-\frac{x^3}{\ell^2} + \frac{2x^2}{\ell} - x\right)}_{s_2} \varphi_1 + \underbrace{\left(-\frac{2x^3}{\ell^3} + \frac{3x^2}{\ell^2}\right)}_{s_3} w_2 + \underbrace{\left(-\frac{x^3}{\ell^2} + \frac{x^2}{\ell}\right)}_{s_4} \varphi_2$$

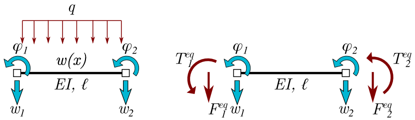
- Work performed by edge forces:

$$W_F = F_1^{\mathrm{eq}} w_1 + T_1^{\mathrm{eq}} \varphi_1 + F_2^{\mathrm{eq}} w_2 + T_2^{\mathrm{eq}} \varphi_2$$

- Work performed by q under the same displacement:

$$W_q = \int_\ell q w(x) \mathrm{d}x = \left(\int_\ell q s_1(x) \mathrm{d}x\right) w_1 + \left(\int_\ell q s_2(x) \mathrm{d}x\right) \varphi_1 + \left(\int_\ell q s_3(x) \mathrm{d}x\right) w_2 + \left(\int_\ell q s_4(x) \mathrm{d}x\right) \varphi_2$$

Work-based element loads



Euler-Bernoulli bending, constant distributed load:

- Displacement field for arbitrary DOFs (ODE with $q = 0$):

$$w(x) = \underbrace{\left(\frac{2x^3}{\ell^3} - \frac{3x^2}{\ell^2} + 1\right)}_{s_1} w_1 + \underbrace{\left(-\frac{x^3}{\ell^2} + \frac{2x^2}{\ell} - x\right)}_{s_2} \varphi_1 + \underbrace{\left(-\frac{2x^3}{\ell^3} + \frac{3x^2}{\ell^2}\right)}_{s_3} w_2 + \underbrace{\left(-\frac{x^3}{\ell^2} + \frac{x^2}{\ell}\right)}_{s_4} \varphi_2$$

- Work performed by edge forces:

$$W_F = F_1^{\mathrm{eq}} w_1 + T_1^{\mathrm{eq}} \varphi_1 + F_2^{\mathrm{eq}} w_2 + T_2^{\mathrm{eq}} \varphi_2$$

- Work performed by q under the same displacement:

$$W_q = \int_\ell q w(x) \mathrm{d}x = \left(\int_\ell q s_1(x) \mathrm{d}x\right) w_1 + \left(\int_\ell q s_2(x) \mathrm{d}x\right) \varphi_1 + \left(\int_\ell q s_3(x) \mathrm{d}x\right) w_2 + \left(\int_\ell q s_4(x) \mathrm{d}x\right) \varphi_2$$

- Enforcing $W_F = W_q$ and isolating terms:

$$\mathbf{f}_{\mathrm{eq}} = \begin{bmatrix} F_1^{\mathrm{eq}} \\ T_1^{\mathrm{eq}} \\ F_2^{\mathrm{eq}} \\ T_2^{\mathrm{eq}} \end{bmatrix} = \begin{bmatrix} \frac{q\ell}{2} \\ -\frac{q\ell^2}{12} \\ \frac{q\ell}{2} \\ \frac{q\ell^2}{12} \end{bmatrix}$$

Dirichlet BCs — Static condensation

Up until now displacement BCs have been simple:

- For a fixed DOF, we can simply discard the corresponding equation
- We did this by striking the corresponding row/column in the final system

Dirichlet BCs — Static condensation

Up until now displacement BCs have been simple:

- For a fixed DOF, we can simply discard the corresponding equation
- We did this by striking the corresponding row/column in the final system

To apply nonzero constraints we can partition the system:

$$\begin{bmatrix} \mathbf{K}_{ff} & \mathbf{K}_{fc} \\ \mathbf{K}_{cf} & \mathbf{K}_{cc} \end{bmatrix} \begin{bmatrix} \mathbf{u}_f \\ \mathbf{u}_c \end{bmatrix} = \begin{bmatrix} \mathbf{f}_f \\ \mathbf{f}_c \end{bmatrix}$$

Dirichlet BCs — Static condensation

Up until now displacement BCs have been simple:

- For a fixed DOF, we can simply discard the corresponding equation
- We did this by striking the corresponding row/column in the final system

To apply nonzero constraints we can partition the system:

$$\begin{bmatrix} \mathbf{K}_{ff} & \mathbf{K}_{fc} \\ \mathbf{K}_{cf} & \mathbf{K}_{cc} \end{bmatrix} \begin{bmatrix} \mathbf{u}_f \\ \mathbf{u}_c \end{bmatrix} = \begin{bmatrix} \mathbf{f}_f \\ \mathbf{f}_c \end{bmatrix}$$

- Solve for the unknown DOFs:

$$\mathbf{K}_{\mathrm{ff}}\mathbf{u}_{\mathrm{f}} + \mathbf{K}_{\mathrm{fc}}\mathbf{u}_{\mathrm{c}} = \mathbf{f}_{\mathrm{f}} \quad \Rightarrow \quad \mathbf{u}_{\mathrm{f}} = \mathbf{K}_{\mathrm{ff}}^{-1}\left(\mathbf{f}_{\mathrm{f}} - \mathbf{K}_{\mathrm{fc}}\mathbf{u}_{\mathrm{c}}\right)$$

Dirichlet BCs — Static condensation

Up until now displacement BCs have been simple:

- For a fixed DOF, we can simply discard the corresponding equation
- We did this by striking the corresponding row/column in the final system

To apply nonzero constraints we can partition the system:

$$\begin{bmatrix} \mathbf{K}_{ff} & \mathbf{K}_{fc} \\ \mathbf{K}_{cf} & \mathbf{K}_{cc} \end{bmatrix} \begin{bmatrix} \mathbf{u}_f \\ \mathbf{u}_c \end{bmatrix} = \begin{bmatrix} \mathbf{f}_f \\ \mathbf{f}_c \end{bmatrix}$$

- Solve for the unknown DOFs:

$$\mathbf{K}_{\mathrm{ff}}\mathbf{u}_{\mathrm{f}} + \mathbf{K}_{\mathrm{fc}}\mathbf{u}_{\mathrm{c}} = \mathbf{f}_{\mathrm{f}} \quad \Rightarrow \quad \mathbf{u}_{\mathrm{f}} = \mathbf{K}_{\mathrm{ff}}^{-1}\left(\mathbf{f}_{\mathrm{f}} - \mathbf{K}_{\mathrm{fc}}\mathbf{u}_{\mathrm{c}}\right)$$

- After solving, we can easily recover support reactions:

$$\mathbf{f}_{\mathrm{c}} = \mathbf{K}_{\mathrm{cf}}\mathbf{u}_{\mathrm{f}} + \mathbf{K}_{\mathrm{cc}}\mathbf{u}_{\mathrm{c}}$$

Dirichlet BCs — Static condensation

Up until now displacement BCs have been simple:

- For a fixed DOF, we can simply discard the corresponding equation
- We did this by striking the corresponding row/column in the final system

To apply nonzero constraints we can partition the system:

$$\begin{bmatrix} \mathbf{K}_{ff} & \mathbf{K}_{fc} \\ \mathbf{K}_{cf} & \mathbf{K}_{cc} \end{bmatrix} \begin{bmatrix} \mathbf{u}_f \\ \mathbf{u}_c \end{bmatrix} = \begin{bmatrix} \mathbf{f}_f \\ \mathbf{f}_c \end{bmatrix}$$

- Solve for the unknown DOFs:

$$\mathbf{K}_{\mathrm{ff}}\mathbf{u}_{\mathrm{f}} + \mathbf{K}_{\mathrm{fc}}\mathbf{u}_{\mathrm{c}} = \mathbf{f}_{\mathrm{f}} \quad \Rightarrow \quad \mathbf{u}_{\mathrm{f}} = \mathbf{K}_{\mathrm{ff}}^{-1}\left(\mathbf{f}_{\mathrm{f}} - \mathbf{K}_{\mathrm{fc}}\mathbf{u}_{\mathrm{c}}\right)$$

- After solving, we can easily recover support reactions:

$$\mathbf{f}_{\mathrm{c}} = \mathbf{K}_{\mathrm{cf}}\mathbf{u}_{\mathrm{f}} + \mathbf{K}_{\mathrm{cc}}\mathbf{u}_{\mathrm{c}}$$

- Note: $\mathbf{f}_{\mathrm{c}}$ includes both nodal loads, equivalent loads and support reactions.

Dirichlet BCs — Size-preserving approach

The approach from before can be annoying to code:

- Reordering the system costs computation time
- Gains when inverting the stiffness matrix are very limited $(N_c \ll N_f)$

Alternatively, we can modify the relevant equations and solve the full system:

- Support reactions recovered later from the unconstrained system

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

Dirichlet BCs — Size-preserving approach

The approach from before can be annoying to code:

- Reordering the system costs computation time
- Gains when inverting the stiffness matrix are very limited ($N_c \ll N_f$)

Alternatively, we can modify the relevant equations and solve the full system:

- Support reactions recovered later from the unconstrained system

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} \\ 0 & 1 & 0 \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} f_1 \\ \Delta_2 \\ f_3 \end{bmatrix}$$

Dirichlet BCs — Size-preserving approach

The approach from before can be annoying to code:

- Reordering the system costs computation time
- Gains when inverting the stiffness matrix are very limited ($N_c \ll N_f$)

Alternatively, we can modify the relevant equations and solve the full system:

- Support reactions recovered later from the unconstrained system

$$\begin{bmatrix} K_{11} & 0 & K_{13} \\ 0 & 1 & 0 \\ K_{31} & 0 & K_{33} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} f_1 - K_{12}\Delta_2 \\ \Delta_2 \\ f_3 - K_{32}\Delta_2 \end{bmatrix}$$

Element-level postprocessing

From discrete nodal displacements to continuum element fields:

- Assemble and solve the global system of equations:

$$\mathbf{K}\mathbf{u} = \mathbf{f}$$

Element-level postprocessing

From discrete nodal displacements to continuum element fields:

- Assemble and solve the global system of equations:

$$\mathbf{K}\mathbf{u} = \mathbf{f}$$

- Select $\mathbf{u}^e$ from $\mathbf{u}$ and go back to the local coordinate system:

$$\overline{\mathbf{u}}^e = \mathbf{T}\mathbf{u}^e$$

Element-level postprocessing

From discrete nodal displacements to continuum element fields:

- Assemble and solve the global system of equations:

$$\mathbf{K}\mathbf{u} = \mathbf{f}$$

- Select $\mathbf{u}^e$ from $\mathbf{u}$ and go back to the local coordinate system:

$$\overline{\mathbf{u}}^e = \mathbf{T}\mathbf{u}^e$$

- From the ODE solution, recover relevant equations as function of $\overline{\mathbf{u}}^e$, e.g.:

$$w(x) = \underbrace{\left(\frac{2x^3}{\ell^3} - \frac{3x^2}{\ell^2} + 1\right)}_{s_1} w_1 + \underbrace{\left(-\frac{x^3}{\ell^2} + \frac{2x^2}{\ell} - x\right)}_{s_2} \varphi_1 + \underbrace{\left(-\frac{2x^3}{\ell^3} + \frac{3x^2}{\ell^2}\right)}_{s_3} w_2 + \underbrace{\left(-\frac{x^3}{\ell^2} + \frac{x^2}{\ell}\right)}_{s_4} \varphi_2$$

- Finally, plot the results! Works for displacements and any other internal field (e.g. moments)

Matrix method versus FEM

The two methods give the same results for bars. However:

- The matrix method solves the strong form ODEs exactly
- FEM solves the weak form problem on the shape function space
- Matrix method: strong form solved locally, elements glued together through equilibrium
- FEM: The weak form is solved globally

Matrix method versus FEM

The two methods give the same results for bars. However:

- The matrix method solves the strong form ODEs exactly
- FEM solves the weak form problem on the shape function space
- Matrix method: strong form solved locally, elements glued together through equilibrium
- FEM: The weak form is solved globally

But how can they give the same solution?

- The "approximation" assumed by FEM (linear shape functions for extension, cubic for bending) turn out to be the exact ODE solution

Matrix method versus FEM

The two methods give the same results for bars. However:

- The matrix method solves the strong form ODEs exactly
- FEM solves the weak form problem on the shape function space
- Matrix method: strong form solved locally, elements glued together through equilibrium
- FEM: The weak form is solved globally

But how can they give the same solution?

- The "approximation" assumed by FEM (linear shape functions for extension, cubic for bending) turn out to be the exact ODE solution

Then why don't we just use the matrix method for everything?

- Gluing elements through equilibrium only works in 1D
- Exact ODE solutions in 2D generally do not exist

Example — 3D frame with torsion

Full solution by hand to demonstrate all steps:

- Definition of a new element (torsion)
- Element reduction for tractability (bending)
- Element loads, support reactions (including distributed loads), postprocessing

Values for numerical calculation:

- $EI = 1000$ kNm^2
- $GI_t = 800$ kNm^2
- $\ell = 2$ m
- $T = 4$ kNm
- $q = 6$ kN/m
- $m = 2$ kNm/m