CIEM5000: Structural Engineering Base

The Matrix Method in Statics

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:

- Recap differential equation for structures
- Degrees of freedom at nodes
- Local and global stiffness matrix
- Neumann and Diriclet boundary conditions
- Local-global transformations
- Example: Displacements of extension bar
- Workshop: Implement and check missing components, and solve a complicated frame

Learning Objectives

At the end of this module, you should be able to:

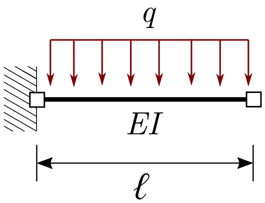
- Translate the main steps of the matrix method into a set of programming classes with distinct tasks
- Extend the classes to solve arbitrarily complex frame problems in statics
- Postprocess the analyses and recover continuum fields exactly

Learning setup:

- Lectures on theoretical aspects ($2 \times 2\,\mathrm{h}$)
- Two guided, non-graded workshops ($2 \times 2\,\mathrm{h}$), solutions provided afterwards
- Additional non-compulsory assignments exercises which you're ready for after the workshops
- Graded assignment as part of report

Recap: A single-field problem

Getting to an ODE:

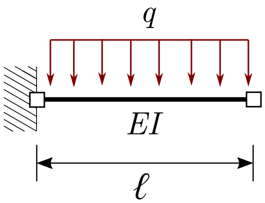


Recap: A single-field problem

Getting to an ODE:

- Kinematic relations:

$$\varphi = -\frac{\mathrm{d}w}{\mathrm{d}x} \quad \kappa = \frac{\mathrm{d}\varphi}{\mathrm{d}x}$$



Recap: A single-field problem

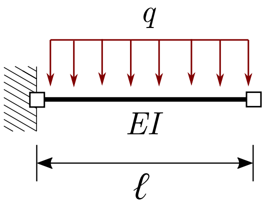
Getting to an ODE:

- Kinematic relations:

$$\varphi = -\frac{\mathrm{d}w}{\mathrm{d}x} \quad \kappa = \frac{\mathrm{d}\varphi}{\mathrm{d}x}$$

- Constitutive relations:

$$M = EI\kappa$$



Recap: A single-field problem

Getting to an ODE:

- Kinematic relations:

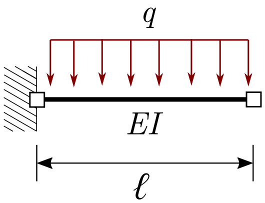
$$\varphi = -\frac{\mathrm{d}w}{\mathrm{d}x} \quad \kappa = \frac{\mathrm{d}\varphi}{\mathrm{d}x}$$

- Constitutive relations:

$$M = EI\kappa$$

- Equilibrium relations:

$$\frac{\mathrm{d}V}{\mathrm{d}x} = -q \quad \frac{\mathrm{d}M}{\mathrm{d}x} = V$$



Recap: A single-field problem

Getting to an ODE:

- Kinematic relations:

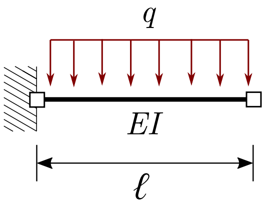
$$\varphi = -\frac{\mathrm{d}w}{\mathrm{d}x} \quad \kappa = \frac{\mathrm{d}\varphi}{\mathrm{d}x}$$

- Constitutive relations:

$$M = EI\kappa$$

- Equilibrium relations:

$$\frac{\mathrm{d}V}{\mathrm{d}x} = -q \quad \frac{\mathrm{d}M}{\mathrm{d}x} = V$$

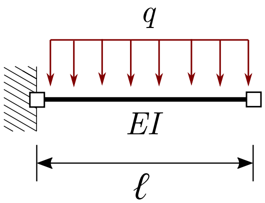


Combining it all into a single differential equation:

$$EI\frac{\mathrm{d}^4w}{\mathrm{d}x^4} = q$$

Recap: A single-field problem

Solving the ODE (strong form!):

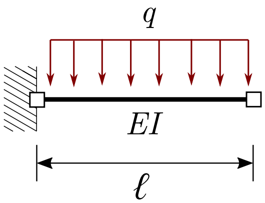


Recap: A single-field problem

Solving the ODE (strong form!):

- Integrate the ODE, exposing integration constants:

$$w(x) = \frac{qx^4}{24EI} + \frac{C_1 x^3}{6} + \frac{C_2 x^2}{2} + C_3 x + C_4$$



Recap: A single-field problem

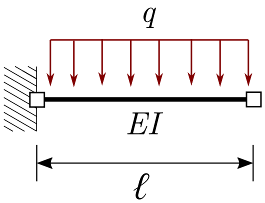
Solving the ODE (strong form!):

- Integrate the ODE, exposing integration constants:

$$w(x) = \frac{qx^4}{24EI} + \frac{C_1x^3}{6} + \frac{C_2x^2}{2} + C_3x + C_4$$

- Enforce boundary conditions:

$$w(0) = 0 \quad \varphi(0) = 0 \quad M(\ell) = 0 \quad V(\ell) = 0$$



Recap: A single-field problem

Solving the ODE (strong form!):

- Integrate the ODE, exposing integration constants:

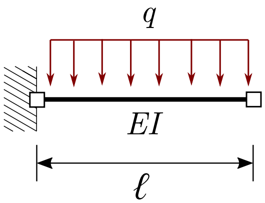
$$w(x) = \frac{qx^4}{24EI} + \frac{C_1x^3}{6} + \frac{C_2x^2}{2} + C_3x + C_4$$

- Enforce boundary conditions:

$$w(0) = 0 \quad \varphi(0) = 0 \quad M(\ell) = 0 \quad V(\ell) = 0$$

- Solve the system for the constants:

$$C_1 = -\frac{q\ell}{EI} \quad C_2 = \frac{q\ell^2}{2EI} \quad C_3 = C_4 = 0$$



Recap: A single-field problem

Solving the ODE (strong form!):

- Integrate the ODE, exposing integration constants:

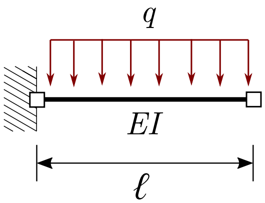
$$w(x) = \frac{qx^4}{24EI} + \frac{C_1x^3}{6} + \frac{C_2x^2}{2} + C_3x + C_4$$

- Enforce boundary conditions:

$$w(0) = 0 \quad \varphi(0) = 0 \quad M(\ell) = 0 \quad V(\ell) = 0$$

- Solve the system for the constants:

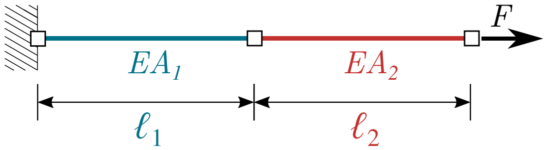
$$C_1 = -\frac{q\ell}{EI} \quad C_2 = \frac{q\ell^2}{2EI} \quad C_3 = C_4 = 0$$



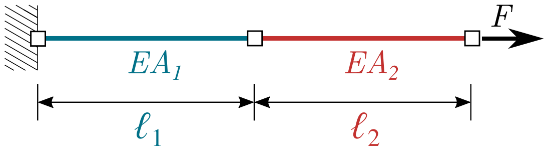
Substituting the constants, a final solution for w can be found:

$$w(x) = \frac{qx^4}{24EI} - \frac{q\ell x^3}{6EI} + \frac{q\ell^2x^2}{4EI}$$

Recap: A two-field problem



Recap: A two-field problem



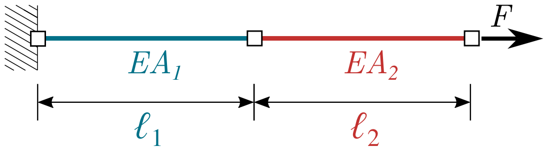
Field 1:

ODE: $EA_1 \dfrac{\mathrm{d}^2 u_1}{\mathrm{d}x^2} = 0$

Field: $u_1 = C_1 x + C_2$

BC: $u_1(0) = 0$

Recap: A two-field problem



Field 1:

ODE: $EA_1 \dfrac{\mathrm{d}^2 u_1}{\mathrm{d}x^2} = 0$

Field: $u_1 = C_1 x + C_2$

BC: $u_1(0) = 0$

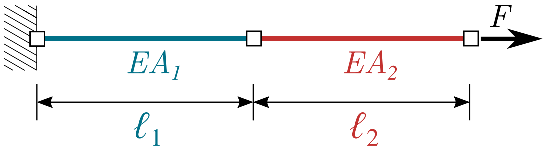
Field 2:

ODE: $EA_2 \dfrac{\mathrm{d}^2 u_2}{\mathrm{d}x^2} = 0$

Field: $u_2 = C_3 x + C_4$

BC: $N_2(\ell_1 + \ell_2) = F$

Recap: A two-field problem



Field 1:

ODE: $EA_1 \dfrac{\mathrm{d}^2 u_1}{\mathrm{d}x^2} = 0$

Field: $u_1 = C_1 x + C_2$

BC: $u_1(0) = 0$

Field 2:

ODE: $EA_2 \dfrac{\mathrm{d}^2 u_2}{\mathrm{d}x^2} = 0$

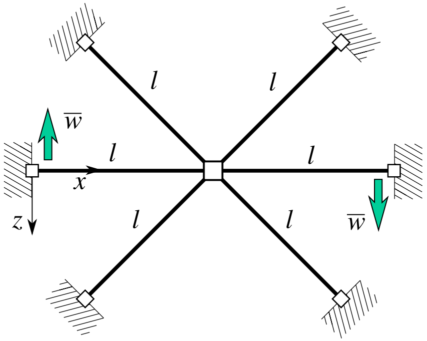
Field: $u_2 = C_3 x + C_4$

BC: $N_2(\ell_1 + \ell_2) = F$

IC: $u_1(\ell_1) = u_2(\ell_1) \quad N_1(\ell_1) = N_2(\ell_1)$

Okay, easy. But how about this one?

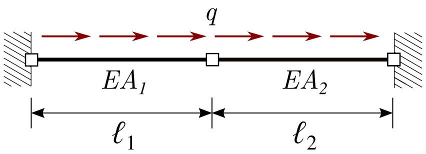
Integration constants? Interface conditions? It gets annoying very quickly...



Remember the displacement method?

Instead of solving for integration constants, we could solve for nodal displacements as we did before for statically indeterminate structures:

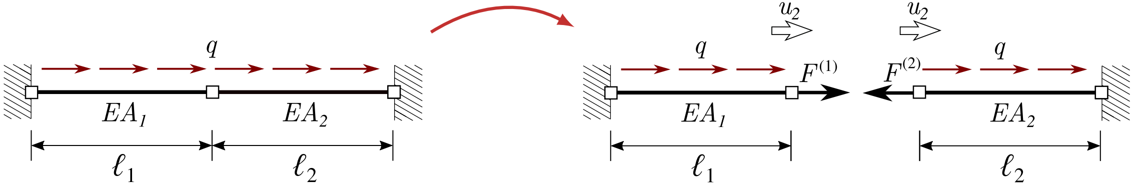
- Chop the structure into statically-determinate parts
- Solve each separately then reinstate equilibrium at the interface



Remember the displacement method?

Instead of solving for integration constants, we could solve for nodal displacements as we did before for statically indeterminate structures:

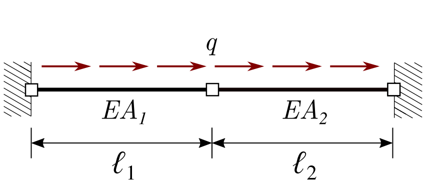
- Chop the structure into statically-determinate parts
- Solve each separately then reinstate equilibrium at the interface

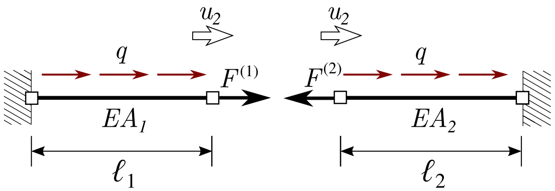


Remember the displacement method?

Instead of solving for integration constants, we could solve for nodal displacements as we did before for statically indeterminate structures:

- Chop the structure into statically-determinate parts
- Solve each separately then reinstate equilibrium at the interface





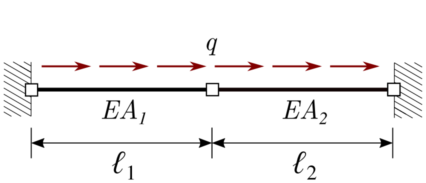
$$u_2 = \frac{\ell_1}{EA_1} F^{(1)} + \frac{\ell_1^2 q}{2EA_1} \qquad u_2 = -\frac{\ell_2}{EA_2} F^{(2)} + \frac{\ell_2^2 q}{2EA_2}$$

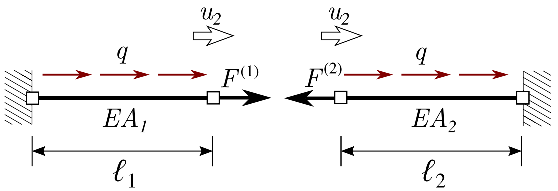
TUDelft

Remember the displacement method?

Instead of solving for integration constants, we could solve for nodal displacements as we did before for statically indeterminate structures:

- Chop the structure into statically-determinate parts
- Solve each separately then reinstate equilibrium at the interface

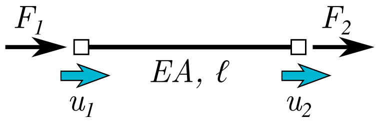




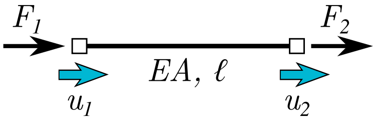
$$u_2 = \frac{\ell_1}{EA_1}F^{(1)} + \frac{\ell_1^2 q}{2EA_1} \qquad u_2 = -\frac{\ell_2}{EA_2}F^{(2)} + \frac{\ell_2^2 q}{2EA_2}$$

$$F^{(1)} = F^{(2)} \quad \Rightarrow \quad u_2 = \frac{\frac{\ell_1 q}{2} + \frac{\ell_2 q}{2}}{\frac{EA_1}{\ell_1} + \frac{EA_2}{\ell_2}}$$

Is there an even more structured way? Deformation of a single element



Is there an even more structured way? Deformation of a single element



ODE solution:

$$EA\frac{\mathrm{d}^2u}{\mathrm{d}x^2} = 0$$

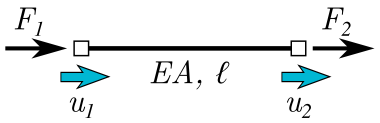
$$u(x) = C_1x + C_2$$

$$u(0) = u_1 \quad u(\ell) = u_2$$

$$C_1 = \frac{u_2 - u_1}{\ell} \quad C_2 = u_1$$

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

Is there an even more structured way? Deformation of a single element



ODE solution:

$$EA\frac{\mathrm{d}^2u}{\mathrm{d}x^2} = 0$$

$$u(x) = C_1x + C_2$$

$$u(0) = u_1 \quad u(\ell) = u_2$$

$$C_1 = \frac{u_2 - u_1}{\ell} \quad C_2 = u_1$$

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

Element forces:

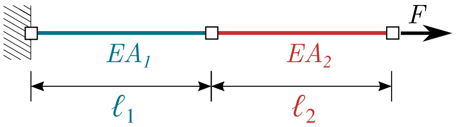
$$N = \frac{EA}{\ell}(u_2 - u_1)$$

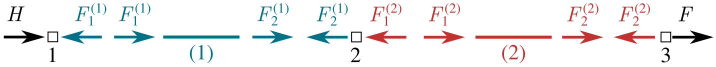
Nodal forces:

$$F_1 = \frac{EA}{\ell}u_1 - \frac{EA}{\ell}u_2$$

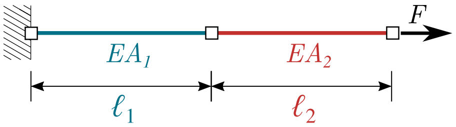
$$F_2 = -\frac{EA}{\ell}u_1 + \frac{EA}{\ell}u_2$$

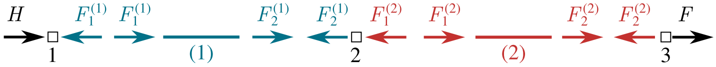
How to combine elements? Nodal equilibrium





How to combine elements? Nodal equilibrium





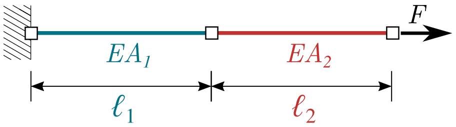
Node equilibrium:

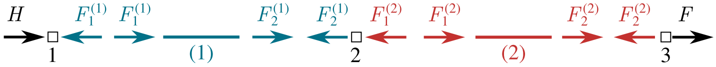
$$\sum F_1 = 0 \Rightarrow -\frac{EA_1}{\ell_1}u_1 + \frac{EA_1}{\ell_1}u_2 + H = 0$$

$$\sum F_2 = 0 \Rightarrow \frac{EA_1}{\ell_1}u_1 - \frac{EA_1}{\ell_1}u_2 - \frac{EA_2}{\ell_2}u_2 + \frac{EA_2}{\ell_2}u_3 = 0$$

$$\sum F_3 = 0 \Rightarrow \frac{EA_2}{\ell_2}u_2 - \frac{EA_2}{\ell_2}u_3 + F = 0$$

How to combine elements? Nodal equilibrium





Node equilibrium:

$$\sum F_1 = 0 \Rightarrow -\frac{EA_1}{\ell_1}u_1 + \frac{EA_1}{\ell_1}u_2 + H = 0$$

$$\sum F_2 = 0 \Rightarrow \frac{EA_1}{\ell_1}u_1 - \frac{EA_1}{\ell_1}u_2 - \frac{EA_2}{\ell_2}u_2 + \frac{EA_2}{\ell_2}u_3 = 0$$

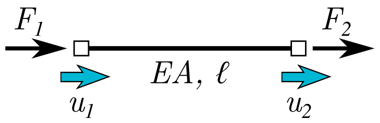
$$\sum F_3 = 0 \Rightarrow \frac{EA_2}{\ell_2}u_2 - \frac{EA_2}{\ell_2}u_3 + F = 0$$

Combining and rearranging:

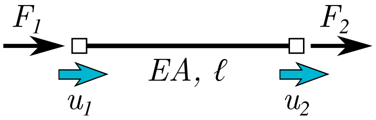
$$-\sum_e \mathbf{f}^e + \mathbf{f}_{\text{nodal}} = \mathbf{0}$$

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

Deformation of a single element – matrix form



Deformation of a single element – matrix form

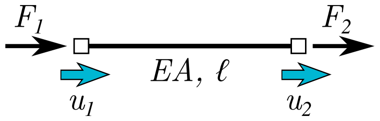


Nodal forces:

$$F_1 = \frac{EA}{\ell}u_1 - \frac{EA}{\ell}u_2$$

$$F_2 = -\frac{EA}{\ell}u_1 + \frac{EA}{\ell}u_2$$

Deformation of a single element – matrix form



Nodal forces:

$$F_1 = \frac{EA}{\ell}u_1 - \frac{EA}{\ell}u_2$$

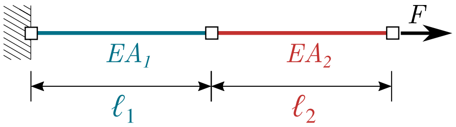
$$F_2 = -\frac{EA}{\ell}u_1 + \frac{EA}{\ell}u_2$$

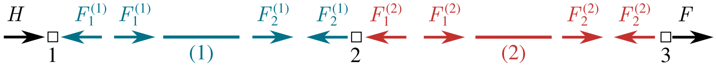
Matrix formulation

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

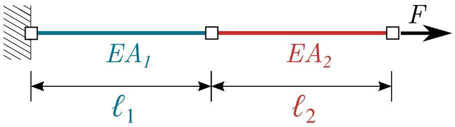
$$\mathbf{K}^{(e)}\mathbf{u}^{(e)} = \mathbf{f}^{(e)}$$

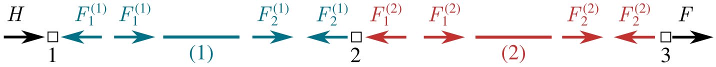
Nodal equilibrium – matrix form





Nodal equilibrium – matrix form





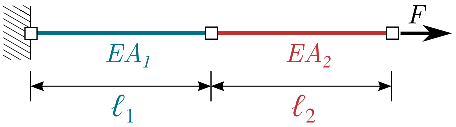
Node equilibrium:

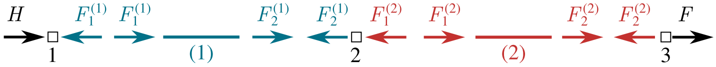
$$\sum F_1 = 0 \Rightarrow -\frac{EA_1}{\ell_1}u_1 + \frac{EA_1}{\ell_1}u_2 + H = 0$$

$$\sum F_2 = 0 \Rightarrow \frac{EA_1}{\ell_1}u_1 - \frac{EA_1}{\ell_1}u_2 - \frac{EA_2}{\ell_2}u_2 + \frac{EA_2}{\ell_2}u_3 = 0$$

$$\sum F_3 = 0 \Rightarrow \frac{EA_2}{\ell_2}u_2 - \frac{EA_2}{\ell_2}u_3 + F = 0$$

Nodal equilibrium – matrix form





Node equilibrium:

$$\sum F_1 = 0 \Rightarrow -\frac{EA_1}{\ell_1}u_1 + \frac{EA_1}{\ell_1}u_2 + H = 0$$

$$\sum F_2 = 0 \Rightarrow \frac{EA_1}{\ell_1}u_1 - \frac{EA_1}{\ell_1}u_2 - \frac{EA_2}{\ell_2}u_2 + \frac{EA_2}{\ell_2}u_3 = 0$$

$$\sum F_3 = 0 \Rightarrow \frac{EA_2}{\ell_2}u_2 - \frac{EA_2}{\ell_2}u_3 + F = 0$$

Matrix formulation:

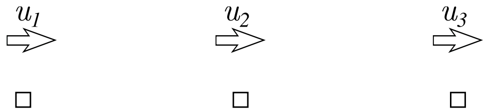
$$\begin{bmatrix} \frac{EA_1}{\ell_1} & -\frac{EA_1}{\ell_1} & 0 \\ -\frac{EA_1}{\ell_1} & \frac{EA_1}{\ell_1} + \frac{EA_2}{\ell_2} & -\frac{EA_2}{\ell_2} \\ 0 & -\frac{EA_2}{\ell_2} & \frac{EA_2}{\ell_2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} H \\ 0 \\ F \end{bmatrix}$$

$$\mathbf{Ku} = \mathbf{f}$$

A more structured way to work

Steps:

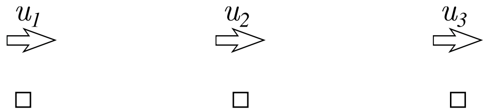
- Identify degrees of freedom at nodes (DOFs)



A more structured way to work

Steps:

- Identify degrees of freedom at nodes (DOFs)
- Initialize the system with zeros

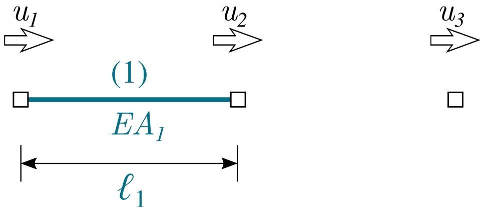


$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

A more structured way to work

Steps:

- Identify degrees of freedom at nodes (DOFs)
- Initialize the system with zeros
- Assemble stiffness, element by element

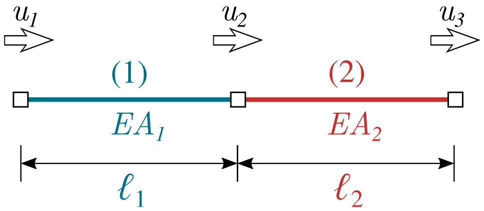


$$\begin{bmatrix} \frac{EA_1}{\ell_1} & -\frac{EA_1}{\ell_1} & 0 \\ -\frac{EA_1}{\ell_1} & \frac{EA_1}{\ell_1} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

A more structured way to work

Steps:

- Identify degrees of freedom at nodes (DOFs)
- Initialize the system with zeros
- Assemble stiffness, element by element

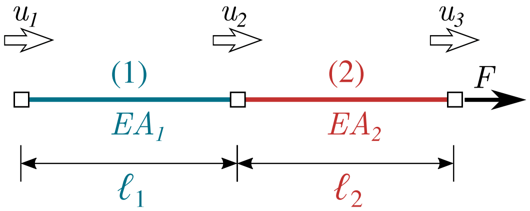


$$\begin{bmatrix} \frac{EA_1}{\ell_1} & -\frac{EA_1}{\ell_1} & 0 \\ -\frac{EA_1}{\ell_1} & \frac{EA_1}{\ell_1} + \frac{EA_2}{\ell_2} & -\frac{EA_2}{\ell_2} \\ 0 & -\frac{EA_2}{\ell_2} & \frac{EA_2}{\ell_2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

A more structured way to work

Steps:

- Identify degrees of freedom at nodes (DOFs)
- Initialize the system with zeros
- Assemble stiffness, element by element
- Apply external loads (Neumann BCs)

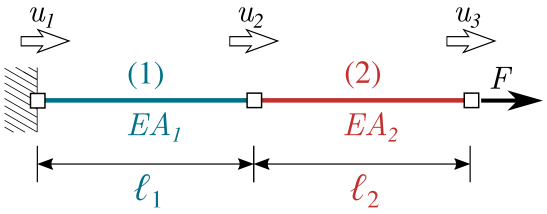


$$\begin{bmatrix} \frac{EA_1}{\ell_1} & -\frac{EA_1}{\ell_1} & 0 \\ -\frac{EA_1}{\ell_1} & \frac{EA_1}{\ell_1} + \frac{EA_2}{\ell_2} & -\frac{EA_2}{\ell_2} \\ 0 & -\frac{EA_2}{\ell_2} & \frac{EA_2}{\ell_2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix}$$

A more structured way to work

Steps:

- Identify degrees of freedom at nodes (DOFs)
- Initialize the system with zeros
- Assemble stiffness, element by element
- Apply external loads (Neumann BCs)
- Apply prescribed displacements (Dirichlet BCs)

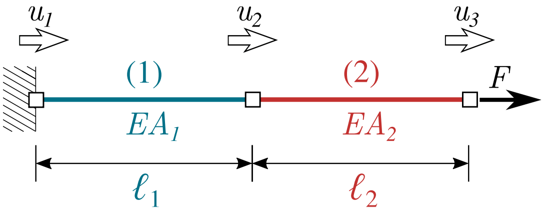


$$\begin{bmatrix} \frac{EA_1}{\ell_1} & -\frac{EA_1}{\ell_1} & 0 \\ -\frac{EA_1}{\ell_1} & \frac{EA_1}{\ell_1}+\frac{EA_2}{\ell_2} & -\frac{EA_2}{\ell_2} \\ 0 & -\frac{EA_2}{\ell_2} & \frac{EA_2}{\ell_2} \end{bmatrix} \begin{bmatrix} 0 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} H \\ 0 \\ F \end{bmatrix}$$

A more structured way to work

Steps:

- Identify degrees of freedom at nodes (DOFs)
- Initialize the system with zeros
- Assemble stiffness, element by element
- Apply external loads (Neumann BCs)
- Apply prescribed displacements (Dirichlet BCs)
- Solve for the unkown nodal displacements



$$\begin{bmatrix} \frac{EA_1}{\ell_1} & -\frac{EA_1}{\ell_1} & 0 \\ -\frac{EA_1}{\ell_1} & \frac{EA_1}{\ell_1} + \frac{EA_2}{\ell_2} & -\frac{EA_2}{\ell_2} \\ 0 & -\frac{EA_2}{\ell_2} & \frac{EA_2}{\ell_2} \end{bmatrix} \begin{bmatrix} 0 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} H \\ 0 \\ F \end{bmatrix}$$

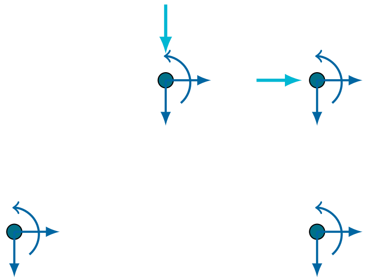
$$\begin{bmatrix} \frac{EA_1}{\ell_1} + \frac{EA_2}{\ell_2} & -\frac{EA_2}{\ell_2} \\ -\frac{EA_2}{\ell_2} & \frac{EA_2}{\ell_2} \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ F \end{bmatrix}$$

$$u_2 = \frac{F\ell_1}{EA_1} \qquad u_3 = \frac{F\left(EA_1\ell_2 + EA_2\ell_1\right)}{EA_1\,EA_2}$$

Coding the matrix method

The method is well structured and can be broken down as follows:

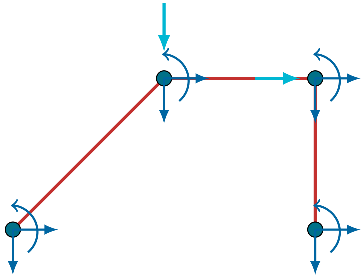
- A list of Nodes floating in space with loads and DOFs associated to them



Coding the matrix method

The method is well structured and can be broken down as follows:

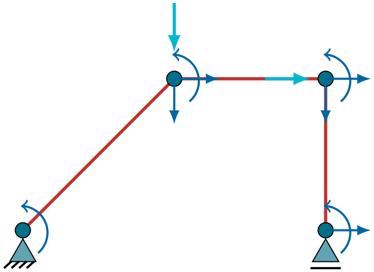
- A list of Nodes floating in space with loads and DOFs associated to them
- A list of Elements defined by linking two nodes together



Coding the matrix method

The method is well structured and can be broken down as follows:

- A list of Nodes floating in space with loads and DOFs associated to them
- A list of Elements defined by linking two nodes together
- A Constrainer to apply Dirichlet boundary conditions

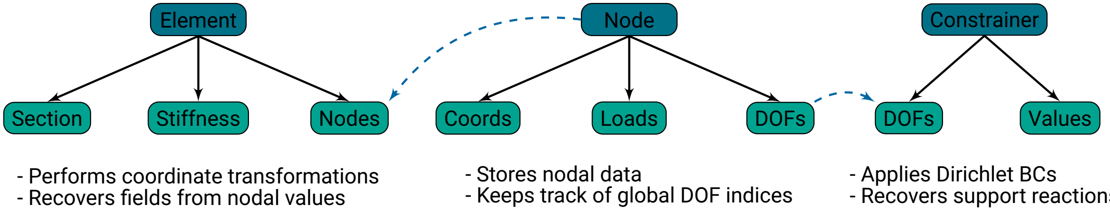


Coding the matrix method

The method is well structured and can be broken down as follows:

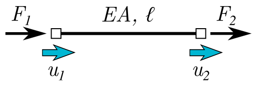
- A list of Nodes floating in space with loads and DOFs associated to them
- A list of Elements defined by linking two nodes together
- A Constrainer to apply Dirichlet boundary conditions

With this in mind, we can define object-oriented code which can be loaded as a python package:

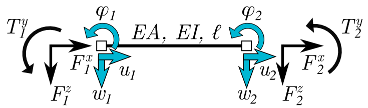

Element
Section
Stiffness
Nodes
Node
Coords
Loads
DOFs
Constrainer
DOFs
Values
- Performs coordinate transformations
- Recovers fields from nodal values
- Stores nodal data
- Keeps track of global DOF indices
- Applies Dirichlet BCs
- Recovers support reactions

Other element types

Different element kinematics and stiffness matrices, same procedure



$$\mathbf{K}^{(e)} = \begin{bmatrix} \frac{EA}{\ell} & -\frac{EA}{\ell} \\ -\frac{EA}{\ell} & \frac{EA}{\ell} \end{bmatrix}$$

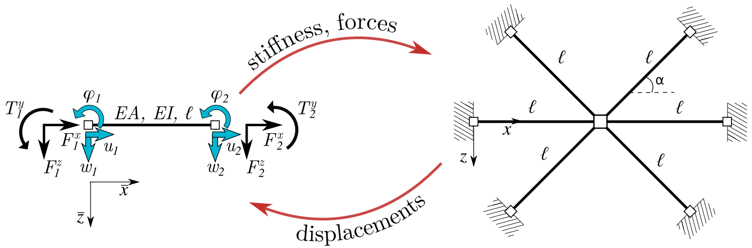


$$\begin{bmatrix} \frac{EA}{\ell} & 0 & 0 & -\frac{EA}{\ell} & 0 & 0 \\ 0 & \frac{12EI}{\ell^3} & -\frac{6EI}{\ell^2} & 0 & -\frac{12EI}{\ell^3} & -\frac{6EI}{\ell^2} \\ 0 & -\frac{6EI}{\ell^2} & \frac{4EI}{\ell} & 0 & \frac{6EI}{\ell^2} & \frac{2EI}{\ell} \\ -\frac{EA}{\ell} & 0 & 0 & \frac{EA}{\ell} & 0 & 0 \\ 0 & -\frac{12EI}{\ell^3} & \frac{6EI}{\ell^2} & 0 & \frac{12EI}{\ell^3} & \frac{6EI}{\ell^2} \\ 0 & -\frac{6EI}{\ell^2} & \frac{2EI}{\ell} & 0 & \frac{6EI}{\ell^2} & \frac{4EI}{\ell} \end{bmatrix}$$

Element orientations, local-global transformations

Defining a local (element) coordinate system is useful:

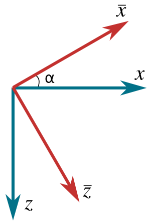
- Single stiffness matrix for every element!
- Assembly: From local to global
- Postprocessing: From global to local



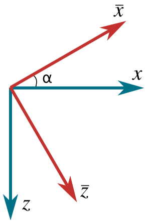
Local-global transformations

Transformations for an arbitrary vector:

$$\begin{bmatrix} v_{\bar{x}} \\ v_{\bar{z}} \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}}_{\mathbf{R}} \begin{bmatrix} v_x \\ v_z \end{bmatrix} \qquad \begin{bmatrix} v_x \\ v_z \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}}_{\mathbf{R}^{\mathrm{T}}} \begin{bmatrix} v_{\bar{x}} \\ v_{\bar{z}} \end{bmatrix}$$



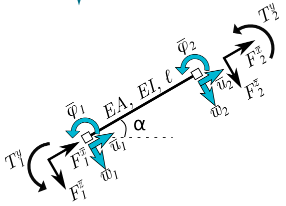
Local-global transformations



Transformations for an arbitrary vector:

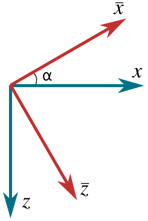
$$\begin{bmatrix} v_{\bar{x}} \\ v_{\bar{z}} \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}}_{\mathbf{R}} \begin{bmatrix} v_x \\ v_z \end{bmatrix} \qquad \begin{bmatrix} v_x \\ v_z \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}}_{\mathbf{R}^\mathrm{T}} \begin{bmatrix} v_{\bar{x}} \\ v_{\bar{z}} \end{bmatrix}$$

Transformations for a complete element:



$$\begin{bmatrix} \bar{u}_1 \\ \bar{w}_1 \\ \bar{\varphi}_1 \\ \bar{u}_2 \\ \bar{w}_2 \\ \bar{\varphi}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & 0 & 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T}} \begin{bmatrix} u_1 \\ w_1 \\ \varphi_1 \\ u_2 \\ w_2 \\ \varphi_2 \end{bmatrix}$$

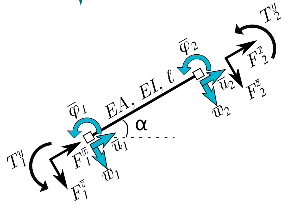
Local-global transformations



Transformations for an arbitrary vector:

$$\begin{bmatrix} v_{\bar{x}} \\ v_{\bar{z}} \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}}_{\mathbf{R}} \begin{bmatrix} v_x \\ v_z \end{bmatrix} \qquad \begin{bmatrix} v_x \\ v_z \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}}_{\mathbf{R}^\mathrm{T}} \begin{bmatrix} v_{\bar{x}} \\ v_{\bar{z}} \end{bmatrix}$$

Transformations for a complete element:



$$\begin{bmatrix} \bar{u}_1 \\ \bar{w}_1 \\ \bar{\varphi}_1 \\ \bar{u}_2 \\ \bar{w}_2 \\ \bar{\varphi}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & 0 & 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T}} \begin{bmatrix} u_1 \\ w_1 \\ \varphi_1 \\ u_2 \\ w_2 \\ \varphi_2 \end{bmatrix}$$

With this we can define the following important transformations:

$$\bar{\mathbf{u}} = \mathbf{T}\mathbf{u} \quad \bar{\mathbf{f}} = \mathbf{T}\mathbf{f} \quad \mathbf{u} = \mathbf{T}^\mathrm{T}\bar{\mathbf{u}} \quad \mathbf{f} = \mathbf{T}^\mathrm{T}\bar{\mathbf{f}}$$

$$\mathbf{K} = \mathbf{T}^\mathrm{T}\bar{\mathbf{K}}\mathbf{T}$$

Outlook

First ungraded workshop:

- Get familiar with an initial Python code
- Implement a few missing parts and perform some sanity checks
- Apply your implementations to a small structure
- Have Git and VS Code with Python, NumPy, matplotlib, SymPy and Jupyter installed and ready
- Never used Git and GitHub? Let me (Tom) know!

Next week:

- One more lecture on theoretical aspects
- Second ungraded workshop to add more implementations and solve a more advanced structure
- Graded assignment: Implement, check and apply new features required for complicated frame structure and additional results.