Bar element: Computation of the global stiffness matrix

1 Geometry

The following terms are defined

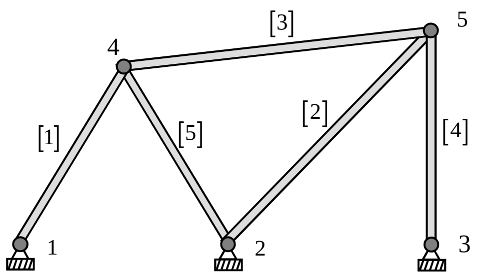
n_d : Problem dimension (e.g. 2 for 2D, 3 for 3D).
n_{el} : Total number of bars.
n : Total number of nodes (joints).
n_{nod} : Number of nodes in a bar, i.e. $n_{nod} = 2$.
n_i : Degrees of freedom per node, i.e. $n_i = n_d$.
n_{dof} : Total number of degrees of freedom, i.e. $n_{dof} = n \times n_i$.

$\mathbf{x}$: Nodal coordinates array ($n \times n_d$):

$$\mathbf{x} = \begin{bmatrix} x_1 & y_1 & z_1 \\ & \vdots & \\ x_n & y_n & z_n \end{bmatrix}.$$

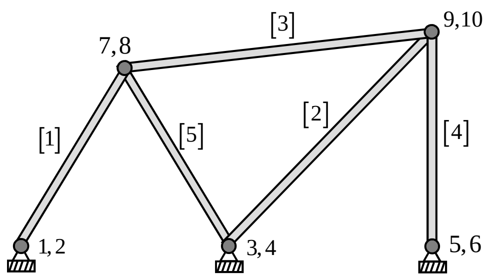
$\mathbf{T}_n$: Nodal connectivity table ($n_{el} \times n_{nod}$). Example:

$$\mathbf{T}_n = \begin{array}{c} \text{Nodes} \\ \begin{array}{cc} a & b \end{array} \\ \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 4 & 5 \\ 3 & 5 \\ 2 & 4 \end{bmatrix} \end{array} \Bigg| \ \text{Elements}$$



$\mathbf{T}_d$: Degrees of freedom connectivity table ($n_{el} \times (n_{nod} \times n_i)$). Example:

$$\mathbf{T}_d = \begin{array}{c} \begin{array}{cc|cc} \multicolumn{2}{c|}{a} & \multicolumn{2}{c}{b} \\ 1 & 2 & 1 & 2 \end{array} \ \begin{array}{l} \text{Nodes} \\ \text{DOFs} \end{array} \\ \begin{bmatrix} 1 & 2 & 7 & 8 \\ 3 & 4 & 9 & 10 \\ 7 & 8 & 9 & 10 \\ 5 & 6 & 9 & 10 \\ 3 & 4 & 7 & 8 \end{bmatrix} \end{array} \Bigg| \ e \ \ \text{Elements}$$



Tip: Given $\mathbf{T}_n$ and n_i, one can algorithmically construct $\mathbf{T}_d$. To do so, a relation between each term $\mathbf{T}_d(e, i)$ with its associated node $\mathbf{T}_n(e, a)$ and degree of freedom j (1… n_i) needs to be found.

2 Computation of the element stiffness matrices

For each bar $e = 1 \dots n_{el}$

a) Compute element stiffness matrix

$$x_1^{(e)} = \mathbf{x}(\mathbf{T}_n(e,1),1), \;\; x_2^{(e)} = \mathbf{x}(\mathbf{T}_n(e,2),1)$$

$$y_1^{(e)} = \mathbf{x}(\mathbf{T}_n(e,1),2), \;\; y_2^{(e)} = \mathbf{x}(\mathbf{T}_n(e,2),2)$$

$$z_1^{(e)} = \mathbf{x}(\mathbf{T}_n(e,1),3), \;\; z_2^{(e)} = \mathbf{x}(\mathbf{T}_n(e,2),3)$$

$$l^{(e)} = \sqrt{\left(x_2^{(e)} - x_1^{(e)}\right)^2 + \left(y_2^{(e)} - y_1^{(e)}\right)^2 + \left(z_2^{(e)} - z_1^{(e)}\right)^2}$$

$$\mathbf{R}^{(e)} = \frac{1}{l^{(e)}} \begin{bmatrix} x_2^{(e)} - x_1^{(e)} & y_2^{(e)} - y_1^{(e)} & z_2^{(e)} - z_1^{(e)} & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2^{(e)} - x_1^{(e)} & y_2^{(e)} - y_1^{(e)} & z_2^{(e)} - z_1^{(e)} \end{bmatrix}$$

$$\mathbf{K}'^{(e)} = \frac{A^{(e)}E^{(e)}}{l^{(e)}} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$\mathbf{K}^{(e)} = \mathbf{R}^{(e)\mathrm{T}} \mathbf{K}'^{(e)} \mathbf{R}^{(e)}$$

b) Store element matrix

For each $r = 1 \dots n_{nod} \times n_i$

For each $s = 1 \dots n_{nod} \times n_i$

$$\mathbf{K}_{el}(r,s,e) = \mathbf{K}^{(e)}(r,s)$$

Next s

Next r

Next bar e

3 Global stiffness matrix assembly

a) Initialization

$$\mathbf{K}_G = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}, \quad \text{dimensions: } n_{dof} \times n_{dof}$$

b) Assembly

$$\mathbf{K}_G = \mathop{\mathbf{A}}_{e=1}^{n_{el}} \mathbf{K}^{(e)}$$

For each bar $e = 1 \dots n_{el}$

For each local degree of freedom $i = 1 \dots n_{nod} \times n_i$ (rows)

$I = \mathbf{T}_d(e, i)$ (corresponding global degree of freedom)

For each local degree of freedom $j = 1 \dots n_{nod} \times n_i$ (columns)

$J = \mathbf{T}_d(e, j)$ (corresponding global degree of freedom)

$\mathbf{K}_G(I, J) = \mathbf{K}_G(I, J) + \mathbf{K}_{el}(i, j, e)$

Next local degree of freedom j

Next local degree of freedom i

Next bar e

4 Global system of equations

a) Global system

$$\mathbf{K}_G \hat{\mathbf{u}} = \widehat{\mathbf{F}}^{ext} + \hat{\mathbf{R}}$$

b) Apply conditions

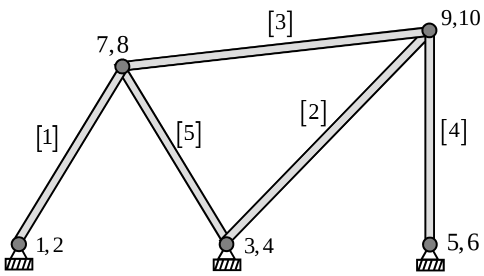
$\nu_R = [\dots]$: Array with the <u>imposed</u> degrees of freedom.

$\nu_L = [\dots]$: Array with the <u>free</u> degrees of freedom.

* In the previous example:

$$\nu_R = [1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6]$$

$$\nu_L = [7 \quad 8 \quad 9 \quad 10]$$



Tip: A relation exists between ν_L and ν_R. In particular, knowing n_{dof} and the imposed degrees of freedom ν_R, then ν_L becomes automatically defined as the remaining degrees of freedom. Matlab's built-in function `setdiff` allows to directly obtain:

$$\nu_L = \texttt{setdiff}(1{:}n_{dof}, \nu_R)$$

c) Partitioned system of equations

$$\begin{bmatrix} \mathbf{K}_{RR} & \mathbf{K}_{RL} \\ \mathbf{K}_{LR} & \mathbf{K}_{LL} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}}_R \\ \hat{\mathbf{u}}_L \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{F}}_R^{ext} \\ \widehat{\mathbf{F}}_L^{ext} \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{R}}_R \\ \mathbf{0} \end{bmatrix}$$

where

$$
\begin{aligned}
\mathbf{K}_{LL} &= \mathbf{K}_G(\nu_L, \nu_L) \\
\mathbf{K}_{LR} &= \mathbf{K}_G(\nu_L, \nu_R) \\
\mathbf{K}_{RL} &= \mathbf{K}_G(\nu_R, \nu_L) \\
\mathbf{K}_{RR} &= \mathbf{K}_G(\nu_R, \nu_R) \\
\widehat{\mathbf{F}}_L^{ext} &= \widehat{\mathbf{F}}^{ext}(\nu_L) \\
\widehat{\mathbf{F}}_R^{ext} &= \widehat{\mathbf{F}}^{ext}(\nu_R)
\end{aligned}
$$

Data:

$\hat{\mathbf{u}}_R$: Imposed displacement vector
$\widehat{\mathbf{F}}^{ext}$: External force vector

Unknowns:

$\hat{\mathbf{u}}_L$: Free displacement vector
$\hat{\mathbf{R}}_R$: Reactions vector

d) System resolution

$$\mathbf{K}_{LL}\hat{\mathbf{u}}_L = \widehat{\mathbf{F}}_L^{ext} - \mathbf{K}_{LR}\hat{\mathbf{u}}_R \quad \rightarrow \quad \hat{\mathbf{u}}_L$$

$$\hat{\mathbf{R}}_R = \mathbf{K}_{RR}\hat{\mathbf{u}}_R + \mathbf{K}_{RL}\hat{\mathbf{u}}_L - \widehat{\mathbf{F}}_R^{ext}$$

e) Obtain generalized displacement vector

$$\hat{\mathbf{u}}(\nu_L, 1) = \hat{\mathbf{u}}_L$$

$$\hat{\mathbf{u}}(\nu_R, 1) = \hat{\mathbf{u}}_R$$

5 Strains and stresses

Strains and stresses for each bar can be computed by means of the following procedure:

For each bar $e = 1 \ldots n_{el}$

a) Compute the rotation matrix

$$x_1^{(e)} = \mathbf{x}(\mathbf{T}_n(e,1),1), \;\; x_2^{(e)} = \mathbf{x}(\mathbf{T}_n(e,2),1)$$

$$y_1^{(e)} = \mathbf{x}(\mathbf{T}_n(e,1),2), \;\; y_2^{(e)} = \mathbf{x}(\mathbf{T}_n(e,2),2)$$

$$z_1^{(e)} = \mathbf{x}(\mathbf{T}_n(e,1),3), \;\; z_2^{(e)} = \mathbf{x}(\mathbf{T}_n(e,2),3)$$

$$l^{(e)} = \sqrt{\left(x_2^{(e)} - x_1^{(e)}\right)^2 + \left(y_2^{(e)} - y_1^{(e)}\right)^2 + \left(z_2^{(e)} - z_1^{(e)}\right)^2}$$

$$\mathbf{R}^{(e)} = \frac{1}{l^{(e)}} \begin{bmatrix} x_2^{(e)} - x_1^{(e)} & y_2^{(e)} - y_1^{(e)} & z_2^{(e)} - z_1^{(e)} & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2^{(e)} - x_1^{(e)} & y_2^{(e)} - y_1^{(e)} & z_2^{(e)} - z_1^{(e)} \end{bmatrix}$$

b) Obtain element displacement in <u>global</u> coordinates

For each local degree of freedom $i = 1 \dots n_{nod} \times n_i$

$$I = \mathbf{T}_d(e, i)$$

$$\hat{\mathbf{u}}^{(e)}(i, 1) = \hat{\mathbf{u}}(I, 1)$$

Next degree of freedom i

c) Compute element displacement in <u>local</u> coordinates

$$\hat{\mathbf{u}}'^{(e)} = \mathbf{R}^{(e)}\hat{\mathbf{u}}^{(e)}$$

d) Compute element strain

$$\hat{\varepsilon}(e, 1) = \frac{1}{l^{(e)}}[-1 \quad 1]\hat{\mathbf{u}}'^{(e)}$$

e) Compute element stress

$$\hat{\sigma}(e, 1) = E^{(e)}\hat{\varepsilon}(e, 1)$$

Next bar e