Appendix D. Beam problems

Problems involving slender structures are solved numerically by splitting the problem into (A) a finite elements one-dimensional beam discretization, and (B) an analysis of the stresses in the two-dimensional cross-sectional area. As an illustrative example consider the beam depicted in Figure D1.

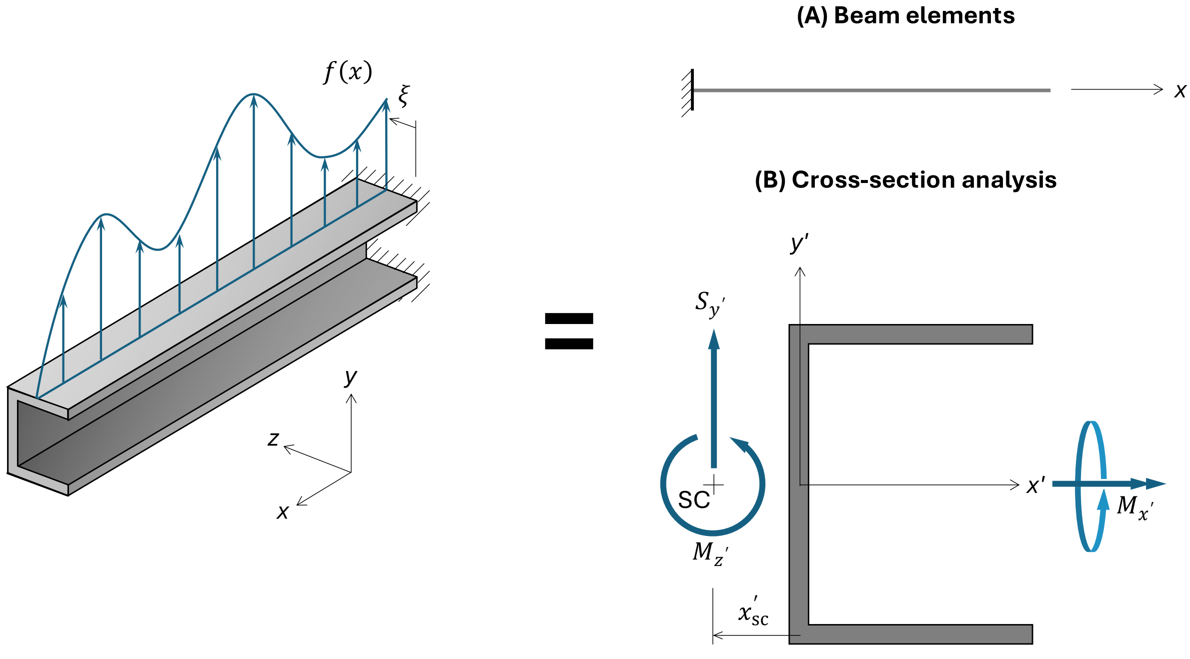

Figure D1. Example beam structure.

For the sake of simplicity, we will consider the problem only involves shear loads in the global vertical y-axis, bending moments about the global z-axis and torsional moments about the longitudinal x-axis. Additionally, the z-axis must coincide with a principal axis (x'-axis) of the cross-section (otherwise cross-moments coupling in both directions should be considered in the bending problem).

***Remark* #D1** *Reference frames*

Notice that the reference axes in the cross-section (x', y', z') do not necessarily coincide with the global beam axes (x, y, z), so special attention must be put into the corresponding signs of the internal loads in each reference frame. In this specific example, we have $Q_y = Q_{y'}$, $M_z = -M_{x'}$, $M_x = M_{z'}$.

The general algorithm to solve such kind of problem involves:

1. Get the cross-section geometrical properties:
    a. For bending: Inertia I with respect to bending axis (z-axis in the global reference frame and x'-axis in the cross-section reference frame).
    b. For torsion: Torsional inertia J and shear center position $\mathbf{x}_{sc}$.
2. Apply the generalized FEM algorithm using the stiffness and force functions specific to beam elements and get the nodal values of the vertical deflection, u_y, bending angle, θ_x, and twist, θ_z.
3. Obtain the internal loads distribution for each element. This allows to get the diagrams of the shear load, Q_y, bending moment, M_z, and torsion moment (about the shear center), M_x.
4. Get the distribution over the cross section of the normal stress, σ_z, tangential stress due to shear loads (applied on the shear centre), τ_x^{S}, τ_y^{S}, and tangential stress due to torsion, τ_x^{T}, τ_y^{T}.

It is worth noting that steps 1 and 4 can be carried out independently just focusing the analysis on the cross-section with generic values of $Q_{y'}$, $M_{x'}$ and $M_{z'}$.

D.1 Analysis of thin-walled cross-sections

The following algorithm can be applied to any open thin-walled cross-section in which an integration path can be determined. For closed cross-sections, first we must open the integration path at any convenient point so the same algorithm can be followed, with specific changes highlighted in **blue**. The first step is to discretize this integration path using small segments numbered sequentially as in Figure D2.

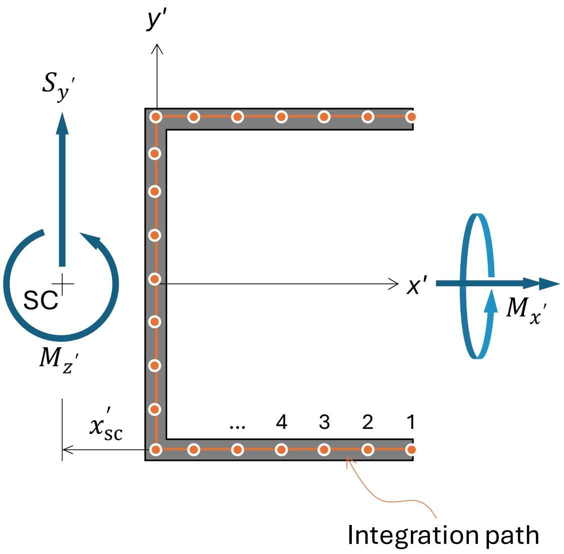

Figure D2. Open thin-walled cross-section discretization.

For that we will need nodal coordinates and nodal connectivities matrices $[\mathbf{x}']$ and $[\mathbf{T}^{\mathrm{n}}]$ providing the position (in the cross-section reference frame) of each node and each segment's nodal indices, respectively. For the algorithm to work as expected, in this case it is important that: (1) the elements are sorted sequentially, i.e. $\mathbf{T}^{\mathrm{n}}(i,1) = \mathbf{T}^{\mathrm{n}}(i-1,2)$, (2) the first node of the first element is at an open end of the integration path.

Additionally, we will define material properties and connectivities matrices $[\mathbf{m}]$ and $[\mathbf{T}^{\mathrm{m}}]$ in which we will include the thickness data of the different elements.

D.1.1 Function to get section properties

Inputs:

- Nodal coordinates matrix of the integration path: $[\mathbf{x}']$
- Nodal connectivities matrix of the integration path: $[\mathbf{T}^{\mathrm{n}}]$
- Material properties matrix containing thicknesses: $[\mathbf{m}]$
- Material connectivities vector linking thicknesses to elements: $\{\mathbf{T}^{\mathrm{m}}\}$

Outputs:

- Centroid of the cross-section: x'_0, y'_0
- Shear center of open cross-section: x'_S, y'_S
- Total area of the cross-section: A_{tot}
- Section inertias about the centroid: $I_{x'x'}, I_{y'y'}, I_{x'y'}$
- Torsional inertia for open section: J
- Enclosed area: A_{in} (only for **closed** cross-sections)

Algorithm:

1. Initialize element length, element area and element centroid vectors:
   $\{\mathbf{l}\} = \{\mathbf{0}\}_{n_{\mathrm{el}}\times 1}$
   $\{\mathbf{A}\} = \{\mathbf{0}\}_{n_{\mathrm{el}}\times 1}$
   $[\hat{\mathbf{x}}'] = [\mathbf{0}]_{n_{\mathrm{el}}\times 2}$
2. Initialize total area and centroid position of the cross-section:
   $A_{\mathrm{tot}} = 0$
   $x_0' = 0$
   $y_0' = 0$
3. In a loop over each segment e from 1 to n_{el}:
   1. Retrieve element's thickness:
      $t = \mathbf{m}(\mathbf{T}^{\mathrm{m}}(e), 1)$
   2. Compute element length, area and centroid:
      $\Delta x' = \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,2),1) - \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,1),1)$
      $\Delta y' = \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,2),2) - \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,1),2)$
      $\mathbf{l}(e) = \sqrt{\Delta x'^2 + \Delta y'^2}$
      $\mathbf{A}(e) = t\mathbf{l}(e)$
      $\hat{\mathbf{x}}'(e,:) = \big(\mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,2),:) + \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,1),:)\big)/2$
   3. Cumulative area and centroid position:
      $A_{\mathrm{tot}} = A_{\mathrm{tot}} + \mathbf{A}(e)$
      $x_0' = x_0' + \mathbf{A}(e)\hat{\mathbf{x}}'(e,1)$
      $y_0' = y_0' + \mathbf{A}(e)\hat{\mathbf{x}}'(e,2)$
4. Compute centroid:
   $x_0' = x_0'/A_{\mathrm{tot}}$
   $y_0' = y_0'/A_{\mathrm{tot}}$
5. Initialize inertias:
   $I_{x'x'} = 0$
   $I_{y'y'} = 0$
   $I_{x'y'} = 0$
   $J = 0$
   $A_{\mathrm{in}} = 0$ (only for **closed** cross-section)
6. In a loop over each segment e from 1 to n_{el}:
   1. Retrieve element's thickness:
      $t = \mathbf{m}(\mathbf{T}^{\mathrm{m}}(e), 1)$
   2. Compute inertia with respect to centroid:
      $\Delta x' = \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,2),1) - \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,1),1)$
      $\Delta y' = \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,2),2) - \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,1),2)$
      $I_{x'x'} = I_{x'x'} + \mathbf{A}(e)\Delta y'^2/12 + \mathbf{A}(e)(\hat{\mathbf{x}}'(e,2) - y_0')^2$
      $I_{y'y'} = I_{y'y'} + \mathbf{A}(e)\Delta x'^2/12 + \mathbf{A}(e)(\hat{\mathbf{x}}'(e,1) - x_0')^2$
      $I_{x'y'} = I_{x'y'} + \mathbf{A}(e)\Delta x'\Delta y'/12 + \mathbf{A}(e)(\hat{\mathbf{x}}'(e,1) - x_0')(\hat{\mathbf{x}}'(e,2) - y_0')$
      For **open** cross-section:
         $J = J + \mathbf{l}(e)t^3/3$
      For **closed** cross-section:
         $A_{\mathrm{in}} = A_{\mathrm{in}} + \|\{\mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,1),1) - x_0', \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,1),2) - y_0', 0\} \times \{\Delta x', \Delta y', 0\}\|/2$
         $J = J + \mathbf{l}(e)/t$
7. Compute torsional inertia (only for **closed** cross-section):
   $J = 4A_{\mathrm{in}}^2/J$

8. Initialize shear flows and shear center position:

$$q_1 = 0$$
$$q_2 = 0$$
$$x'_S = x'_0$$
$$y'_S = y'_0$$

9. In a loop over each segment e from 1 to n_{el}:
    1. Retrieve element's thickness:
    $$t = \mathbf{m}(\mathbf{T}^{\text{m}}(e), 1)$$
    2. Compute coefficients:
    $$\Delta x' = \mathbf{x}'(\mathbf{T}^{\text{n}}(e,2),1) - \mathbf{x}'(\mathbf{T}^{\text{n}}(e,1),1)$$
    $$\Delta y' = \mathbf{x}'(\mathbf{T}^{\text{n}}(e,2),2) - \mathbf{x}'(\mathbf{T}^{\text{n}}(e,1),2)$$
    $$A_1 = \left(I_{y'y'}\Delta y'/2 - I_{x'y'}\Delta x'/2\right)/\left(I_{x'x'}I_{y'y'} - I_{x'y'}^2\right)$$
    $$B_1 = \left(I_{y'y'}(\mathbf{x}'(\mathbf{T}^{\text{n}}(e,1),2) - y'_0) - I_{x'y'}(\mathbf{x}'(\mathbf{T}^{\text{n}}(e,1),1) - x'_0)\right)/\left(I_{x'x'}I_{y'y'} - I_{x'y'}^2\right)$$
    $$A_2 = \left(I_{x'x'}\Delta x'/2 - I_{x'y'}\Delta y'/2\right)/\left(I_{x'x'}I_{y'y'} - I_{x'y'}^2\right)$$
    $$B_2 = \left(I_{x'x'}(\mathbf{x}'(\mathbf{T}^{\text{n}}(e,1),1) - x'_0) - I_{x'y'}(\mathbf{x}'(\mathbf{T}^{\text{n}}(e,1),2) - y'_0)\right)/\left(I_{x'x'}I_{y'y'} - I_{x'y'}^2\right)$$
    $$C = (\mathbf{x}'(\mathbf{T}^{\text{n}}(e,1),1) - x'_0)\Delta y' - (\mathbf{x}'(\mathbf{T}^{\text{n}}(e,1),2) - y'_0)\Delta x'$$
    3. Add moment contribution:
    $$x'_S = x'_S + C\left(q_1 - t\mathbf{l}(e)(A_1/3 + B_1/2)\right)$$
    $$y'_S = y'_S + C\left(q_2 + t\mathbf{l}(e)(A_2/3 + B_2/2)\right)$$
    4. Update shear flows:
    $$q_1 = q_1 - t\mathbf{l}(e)(A_1 + B_1)$$
    $$q_2 = q_2 + t\mathbf{l}(e)(A_2 + B_2)$$

D.1.2 Function to get normal stress distribution

Inputs:

- Nodal coordinates matrix of the integration path: $[\mathbf{x}']$
- Nodal connectivities matrix of the integration path: $[\mathbf{T}^{\text{n}}]$
- Centroid of the cross-section: x'_0, y'_0
- Section inertias about the centroid: $I_{x'x'}, I_{y'y'}, I_{x'y'}$
- Bending moments: $M_{x'}, M_{y'}$

Outputs:

- Normal stress at each element's nodes matrix: $[\boldsymbol{\sigma}]$
- Arc-length positions matrix: $[\mathbf{s}]$ (useful for plot σ vs. s)

Algorithm:

1. Initialize vectors:
$$[\mathbf{s}] = [\mathbf{0}]_{2\times n_{\text{el}}}$$
$$[\boldsymbol{\sigma}] = [\mathbf{0}]_{2\times n_{\text{el}}}$$
2. Compute equivalent inertias and moments:
$$I_x = I_{x'x'} - I_{x'y'}^2/I_{y'y'}$$
$$I_y = I_{y'y'} - I_{x'y'}^2/I_{x'x'}$$
$$M_x = M_{x'} + M_{y'}I_{x'y'}/I_{y'y'}$$
$$M_y = M_{y'} + M_{x'}I_{x'y'}/I_{x'x'}$$
3. In a loop over each segment e from 1 to n_{el}:
    1. Compute element length:

$\Delta x' = \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,2),1) - \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,1),1)$
$\Delta y' = \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,2),2) - \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,1),2)$
$l = \sqrt{\Delta x'^2 + \Delta y'^2}$

2. Compute arc-length position corresponding to second node:
   $\mathbf{s}(1,e) = \mathbf{s}(2,e-1)$ (only for $e > 1$)
   $\mathbf{s}(2,e) = \mathbf{s}(1,e) + l$
3. Compute stress corresponding to second node:
   $\boldsymbol{\sigma}(1,e) = (\mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,1),2) - y_0')M_x/I_x - (\mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,1),1) - x_0')M_y/I_y$
   $\boldsymbol{\sigma}(2,e) = (\mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,2),2) - y_0')M_x/I_x - (\mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,2),1) - x_0')M_y/I_y$

D.1.3 Function to get tangential stress distribution due to shear

Inputs:

- Nodal coordinates matrix of the integration path: $[\mathbf{x}']$
- Nodal connectivities matrix of the integration path: $[\mathbf{T}^{\mathrm{n}}]$
- Material properties matrix containing thicknesses: $[\mathbf{m}]$
- Material connectivities vector linking thicknesses to elements: $\{\mathbf{T}^{\mathrm{m}}\}$
- Centroid of the cross-section: x_0', y_0'
- Section inertias about the centroid: $I_{x'x'}, I_{y'y'}, I_{x'y'}$
- Shear loads: $S_{x'}, S_{y'}$
- Shear center of (open) cross-section: x_S', y_S' (only needed for **closed** cross-sections)
- Enclosed area: A_{in} (only for **closed** cross-sections)

Outputs:

- Tangential stress due to shear vector at each element's nodes matrix: $[\boldsymbol{\tau}^{\mathrm{S}}]$
- Arc-length positions matrix: $[\mathbf{s}]$ (useful for plot τ^{S} vs. s)

Algorithm:

1. Initialize vectors:
   $[\mathbf{s}] = [\mathbf{0}]_{2\times n_{\mathrm{el}}}$
   $[\boldsymbol{\tau}^{\mathrm{S}}] = [\mathbf{0}]_{2\times n_{\mathrm{el}}}$
2. Initialize shear flow:
   For **open** cross-section:
   $q = 0$
   For **closed** cross-section (assuming the centroid is approximately the shear center):
   $q = \left(S_{y'}(x_0' - x_S') - S_{x'}(y_0' - y_S')\right)/2A_{\mathrm{in}}$
3. Compute equivalent inertias and shear:
   $I_x = I_{x'x'} - I_{x'y'}^2/I_{y'y'}$
   $I_y = I_{y'y'} - I_{x'y'}^2/I_{x'x'}$
   $S_x = S_{x'} - S_{y'}I_{x'y'}/I_{x'x'}$
   $S_y = S_{y'} - S_{x'}I_{x'y'}/I_{y'y'}$
4. In a loop over each segment e from 1 to n_{el}:
   1. Retrieve element's thickness:
      $t = \mathbf{m}(\mathbf{T}^{\mathrm{m}}(e),1)$
   2. Compute element length:
      $\Delta x' = \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,2),1) - \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,1),1)$
      $\Delta y' = \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,2),2) - \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e,1),2)$
      $l = \sqrt{\Delta x'^2 + \Delta y'^2}$

3. Compute arc-length position corresponding to second node:
   $\mathbf{s}(1, e) = \mathbf{s}(2, e - 1)$ (only for $e > 1$)
   $\mathbf{s}(2, e) = \mathbf{s}(1, e) + l$
4. Compute shear stress corresponding to first node:
   $\boldsymbol{\tau}^{\mathrm{S}}(1, e) = q/t$
5. Update shear flow:
   $q = q - S_x tl(\Delta x'/2 + \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e, 1), 1) - x'_0)/I_y - S_y tl(\Delta y'/2 + \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e, 1), 2) - y'_0)/I_x$
6. Compute shear stress corresponding to second node:
   $\boldsymbol{\tau}^{\mathrm{S}}(2, e) = q/t$

D.1.4 Function to get tangential stress distribution due to torsion

Inputs:

- Nodal coordinates matrix of the integration path: $[\mathbf{x}']$
- Nodal connectivities matrix of the integration path: $[\mathbf{T}^{\mathrm{n}}]$
- Material properties matrix containing thicknesses: $[\mathbf{m}]$
- Material connectivities vector linking thicknesses to elements: $\{\mathbf{T}^{\mathrm{m}}\}$
- Torsion moment about the shear center: $M_{z'}$
- Torsional inertia: J (only needed for **open** cross-section)
- Enclosed area: A_{in} (only for **closed** cross-section)

Outputs:

- Tangential stress due to shear vector at each element's nodes matrix: $[\boldsymbol{\tau}^{\mathrm{T}}]$
- Arc-length positions matrix: $[\mathbf{s}]$ (useful for plot τ^{T} vs. s)

Algorithm:

1. Initialize vectors:
   $[\mathbf{s}] = [\mathbf{0}]_{2\times n_{\mathrm{el}}}$
   $[\boldsymbol{\tau}^{\mathrm{T}}] = [\mathbf{0}]_{2\times n_{\mathrm{el}}}$
2. In a loop over each segment e from 1 to n_{el}:
   1. Retrieve element's thickness:
      $t = \mathbf{m}(\mathbf{T}^{\mathrm{m}}(e), 1)$
   2. Compute element length:
      $\Delta x' = \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e, 2), 1) - \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e, 1), 1)$
      $\Delta y' = \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e, 2), 2) - \mathbf{x}'(\mathbf{T}^{\mathrm{n}}(e, 1), 2)$
      $l = \sqrt{\Delta x'^2 + \Delta y'^2}$
   3. Compute arc-length position corresponding to second node:
      $\mathbf{s}(1, e) = \mathbf{s}(2, e - 1)$ (only for $e > 1$)
      $\mathbf{s}(2, e) = \mathbf{s}(1, e) + l$
   4. Compute shear stress corresponding to element's nodes:
      For **open** cross-section:
         $\boldsymbol{\tau}^{\mathrm{T}}(1, e) = M_{z'} t/J$
         $\boldsymbol{\tau}^{\mathrm{T}}(2, e) = M_{z'} t/J$
      For **closed** cross-section:
         $\boldsymbol{\tau}^{\mathrm{T}}(1, e) = M_{z'}/2A_{\mathrm{in}} t$
         $\boldsymbol{\tau}^{\mathrm{T}}(2, e) = M_{z'}/2A_{\mathrm{in}} t$

D.2 Element stiffness function

For this function, another set of matrices $[\mathbf{x}]$, $[\mathbf{T}^{\mathrm{n}}]$, $[\mathbf{m}]$ and $[\mathbf{T}^{\mathrm{m}}]$ are required. These are different from the ones used in the cross-section analysis and correspond to the discretization (meshing) of the one-dimensional beam structure in its longitudinal direction. This means also that now our reference system (x, y, z) is the global one (see Figure D1). The material properties needed are: (1) Young's modulus E, (2) Shear modulus G, (3) Bending inertia I, and (4) Torsional inertia J of the cross-section. In this case, I and J are obtained from the cross-section analysis as described in Section D.1. It is worth noting that the beam model implemented here assumes that the inertia I is <u>about a principal axis of the cross-section</u>.

Inputs:

- Element index: e
- Nodal coordinates matrix: $[\mathbf{x}]$
- Nodal connectivities matrix: $[\mathbf{T}^{\mathrm{n}}]$
- Material properties matrix: $[\mathbf{m}]$
- Material connectivities vector: $\{\mathbf{T}^{\mathrm{m}}\}$

Outputs:

- Element stiffness matrix: $[\mathbf{K}^*]$

Algorithm:

1. Get beam length:
   $l = |\mathbf{x}(\mathbf{T}^{\mathrm{n}}(e,2),1) - \mathbf{x}(\mathbf{T}^{\mathrm{n}}(e,1),1)|$
2. Retrieve element's material properties:
   $E = \mathbf{m}(\mathbf{T}^{\mathrm{m}}(e),1)$
   $G = \mathbf{m}(\mathbf{T}^{\mathrm{m}}(e),2)$
   $I = \mathbf{m}(\mathbf{T}^{\mathrm{m}}(e),3)$
   $J = \mathbf{m}(\mathbf{T}^{\mathrm{m}}(e),4)$
3. Compute element stiffness matrix for bending:
$$[\mathbf{K}_{\mathrm{b}}] = \frac{EI}{l^3}\begin{bmatrix} 12 & 6l & 0 & -12 & 6l & 0 \\ 6l & 4l^2 & 0 & -6l & 2l^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -12 & -6l & 0 & 12 & -6l & 0 \\ 6l & 2l^2 & 0 & -6l & 4l^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$
4. Compute element stiffness matrix for torsion:
$$[\mathbf{K}_{\mathrm{t}}] = \frac{GJ}{l}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$
5. Compute total element stiffness matrix:
   $[\mathbf{K}^*] = [\mathbf{K}_{\mathrm{b}}] + [\mathbf{K}_{\mathrm{t}}]$

D.3 Element force function

The algorithm for this function will take into account that a uniformly distributed vertical force $\bar{f}_i$ (in the global y-direction) is applied at a distance ξ in the global z-direction. For an arbitrary distribution along the beam (as depicted in Figure D3), this approximation will be better the smaller the size of the element is.

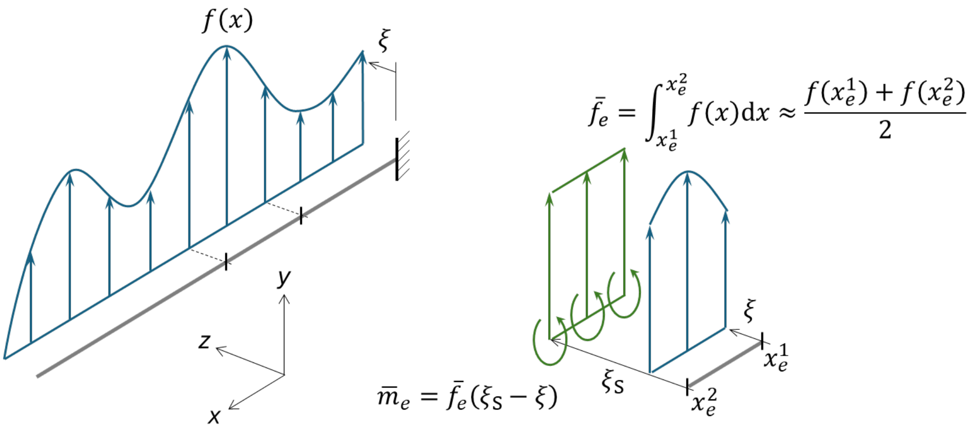

Figure D3. Equivalent uniform distributed force and torsion moment in a beam element.

Inputs:

- Element index: e
- Nodal coordinates matrix: $[\mathbf{x}]$
- Nodal connectivities matrix: $[\mathbf{T}^{\mathrm{n}}]$
- Equivalent vertical force on the element: $\bar{f}_e$
- Equivalent torsion moment on the element: $\bar{m}_e$

Outputs:

- Element force vector: $\{\mathbf{f}^*\}$

Algorithm:

1. Get beam length:
   $l = |\mathbf{x}(\mathbf{T}^{\mathrm{n}}(e,2),1) - \mathbf{x}(\mathbf{T}^{\mathrm{n}}(e,1),1)|$
2. Contribution of distributed shear load:
   $$\{\mathbf{f}_{\mathrm{b}}\} = \bar{f}_e l \begin{Bmatrix} 1/2 \\ l/12 \\ 0 \\ 1/2 \\ -l/12 \\ 0 \end{Bmatrix}$$
3. Contribution of distributed torsion:
   $$\{\mathbf{f}_{\mathrm{t}}\} = \bar{m}_e l \begin{Bmatrix} 0 \\ 0 \\ 1/2 \\ 0 \\ 0 \\ 1/2 \end{Bmatrix}$$
4. Compute total element force vector:
   $\{\mathbf{f}^*\} = \{\mathbf{f}_{\mathrm{b}}\} + \{\mathbf{f}_{\mathrm{t}}\}$

D.4 Element internal forces function

This function allows to compute the loads diagrams along the beam. The output returns the shear load, bending moment and torsion moment about the shear center at the two nodes of each element. This data can then be used to get the corresponding stresses distributions on the cross-section (using the functions described in Sections D.1.2, D.1.3 and D.1.4.

Inputs:

- Nodal coordinates matrix: $[\mathbf{x}]$
- Nodal connectivities matrix: $[\mathbf{T}^{\mathrm{n}}]$
- Degrees of freedom connectivities: $[\mathbf{T}^{\mathrm{d}}]$
- Element stiffness matrices: $[\mathbf{K}^{\mathrm{el}}]$
- Solution degrees of freedom vector: $\{\mathbf{u}\}$

Outputs:

- Element nodal coordinates matrix: $[\mathbf{x}^{\mathrm{el}}]$ (useful for plot $F_{\mathrm{int}}^{\mathrm{el}}$ vs. x^{el})
- Element internal shear: $[\mathbf{S}^{\mathrm{el}}]$
- Element internal bending moment: $[\mathbf{M}_{\mathrm{b}}^{\mathrm{el}}]$
- Element internal torsion moment: $[\mathbf{M}_{\mathrm{t}}^{\mathrm{el}}]$

Algorithm:

1. Initialize vectors:
   $[\mathbf{x}^{\mathrm{el}}] = [\mathbf{0}]_{2\times n_{\mathrm{el}}}$
   $[\mathbf{S}^{\mathrm{el}}] = [\mathbf{0}]_{2\times n_{\mathrm{el}}}$
   $[\mathbf{M}_{\mathrm{b}}^{\mathrm{el}}] = [\mathbf{0}]_{2\times n_{\mathrm{el}}}$
   $[\mathbf{M}_{\mathrm{t}}^{\mathrm{el}}] = [\mathbf{0}]_{2\times n_{\mathrm{el}}}$
2. In a loop over each element e from 1 to n_{el}:
   1. Retrieve element's vertices coordinates:
      $\mathbf{x}^{\mathrm{el}}(e,:) = \mathbf{x}(\mathbf{T}^{\mathrm{n}}(e,:))$
   2. Initialize the element's vertices DOFs vector:
      $\{\mathbf{u}^{\mathrm{el}}\} = \{\mathbf{0}\}_{(n_{\mathrm{ne}}\cdot n_{\mathrm{i}})\times 1}$
   3. In a loop over each element DOF i (from 1 to $n_{\mathrm{ne}} \times n_{\mathrm{i}}$):
      1. Assign the corresponding DOF value:
         $\mathbf{u}^{\mathrm{el}}(i) = \mathbf{u}\left(\mathbf{T}^{\mathrm{d}}(e,i)\right)$
   4. Compute internal forces:
      $\{\mathbf{f}_{\mathrm{int}}^{\mathrm{el}}\} = [\mathbf{K}^{\mathrm{el}}(:,:,e)]\{\mathbf{u}^{\mathrm{el}}\}$
   5. Assign cross-section loads:
      $\mathbf{S}^{\mathrm{el}}(1,e) = -\mathbf{f}_{\mathrm{int}}^{\mathrm{el}}(1)$
      $\mathbf{S}^{\mathrm{el}}(2,e) = \mathbf{f}_{\mathrm{int}}^{\mathrm{el}}(4)$
      $\mathbf{M}_{\mathrm{b}}^{\mathrm{el}}(1,e) = -\mathbf{f}_{\mathrm{int}}^{\mathrm{el}}(2)$
      $\mathbf{M}_{\mathrm{b}}^{\mathrm{el}}(2,e) = \mathbf{f}_{\mathrm{int}}^{\mathrm{el}}(5)$
      $\mathbf{M}_{\mathrm{t}}^{\mathrm{el}}(1,e) = -\mathbf{f}_{\mathrm{int}}^{\mathrm{el}}(3)$
      $\mathbf{M}_{\mathrm{t}}^{\mathrm{el}}(2,e) = \mathbf{f}_{\mathrm{int}}^{\mathrm{el}}(6)$

Notice: Beware of the signs and the potential different reference frames when translating these loads with the cross-section functions inputs to get the corresponding stresses distributions.