Algorithm for problems involving beam elements

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

Algorithm for problems involving beams

1) Input data:

1.1 Cross-section data and materials. For each <u>different</u> cross-section (m):

- Young's modulus: $E^{(m)}$
- Shear modulus: $G^{(m)}$ (*) Alternatively, Poisson ratio $\nu^{(m)}$, $G^{(m)} = E^{(m)}/\left(2\left(1+\nu^{(m)}\right)\right)$
- Density: $\rho^{(m)}$
- Orientation of y' axis: $\hat{\boldsymbol{j}}'^{(m)}$
- Area: $A^{(m)}$
- Polar inertia: $J^{(m)}$
- Area inertias: $I_{y'}^{(m)}, I_{z'}^{(m)}$
- Shear and torsion correction factors: $k_y^{(m)}, k_z^{(m)}, k_\mathrm{t}^{(m)}$

Additionally, for sections with no symmetries or isotropic material properties:

- Cross area inertia: $I_{y'z'}^{(m)}$ (when y' and z' are not principal axes)
- Shear centre $(y_\mathrm{c}'^{(m)}, z_\mathrm{c}'^{(m)})$, neutral centre $(y_0'^{(m)}, z_0'^{(m)})$ and centre of mass $(y_\mathrm{cm}'^{(m)}, z_\mathrm{cm}'^{(m)})$
- Radii of gyration about local axes for mass inertias: $r_{x'}^{(m)}, r_{y'}^{(m)}, r_{z'}^{(m)}, r_{y'z'}^{(m)}$

Algorithm for problems involving beams

1.2 Mesh (discretization) data:

- Nodal coordinates matrix:

$$[\mathbf{X}] = \begin{bmatrix} & \vdots & \\ \hat{x}^{(n)} & \hat{y}^{(n)} & \hat{z}^{(n)} \\ & \vdots & \end{bmatrix} \Bigg\} \text{ Number of nodes } N$$

$\hat{x}^{(n)}$: x-coordinate of node n
$\hat{y}^{(n)}$: y-coordinate of node n
$\hat{z}^{(n)}$: z-coordinate of node n

- Nodal connectivities matrix:

$$[\mathbf{T}_\mathrm{n}] = \begin{bmatrix} & \vdots & \\ n_1^{(e)} & & n_2^{(e)} \\ & \vdots & \end{bmatrix} \Bigg\} \text{ Number of elements } N_e$$

$n_i^{(e)}$: global node # assigned to i-th node in element (e)

- Material connectivites matrix:

$$[\mathbf{T}_\mathrm{m}] = \begin{bmatrix} \vdots \\ m^{(e)} \\ \vdots \end{bmatrix} \Bigg\} \text{ Number of elements } N_e$$

$m^{(e)}$: material index # assigned to element (e)

1.3 Boundary conditions:

- Fix nodes and DOFs matrix:

$$[\mathbf{U}_\mathrm{p}] = \begin{bmatrix} & \vdots & \\ u^{(p)} & n^{(p)} & j^{(p)} \\ & \vdots & \end{bmatrix} \Bigg\} \text{ Number of prescribed DOFs}$$

$u^{(p)}$: value of prescribed displ./rot. (p)
$n^{(p)}$: global node # assigned to (p)
$j^{(p)}$: degree of freedom assigned to (p)

Algorithm for problems involving beams

1.4 External forces:

- Non-null point forces matrix (N):

$$[\mathbf{F}_\mathrm{e}] = \left[\begin{matrix} & \vdots & \\ f^{(q)} & n^{(q)} & j^{(q)} \\ & \vdots & \end{matrix}\right] \left.\vphantom{\begin{matrix} \\ \\ \\ \end{matrix}}\right\} \text{Number of point forces}$$

$f^{(q)}$: value of point force/moment (q)
$n^{(q)}$: global node # assigned to (q)
$j^{(q)}$: degree of freedom assigned to (q)

- Distributed loads matrix (N/m):

$$[\mathbf{Q}_\mathrm{e}] = \left[\begin{matrix} & \vdots & \\ q^{(r)} & n^{(r)} & j^{(r)} \\ & \vdots & \end{matrix}\right] \left.\vphantom{\begin{matrix} \\ \\ \\ \end{matrix}}\right\} \text{Number of DOFs with distributed loads}$$

$q^{(r)}$: value of distr. force/moment (r)
$n^{(r)}$: global node # assigned to (r)
$j^{(r)}$: degree of freedom assigned to (r)

- Body forces matrix (N/kg):

$$[\mathbf{B}_\mathrm{e}] = \left[\begin{matrix} & \vdots & \\ b^{(s)} & n^{(s)} & j^{(s)} \\ & \vdots & \end{matrix}\right] \left.\vphantom{\begin{matrix} \\ \\ \\ \end{matrix}}\right\} \text{Number of DOFs with body forces}$$

$b^{(s)}$: value of body force/moment (s)
$n^{(s)}$: global node # assigned to (s)
$j^{(s)}$: degree of freedom assigned to (s)

<u>Note</u>: Recall that degrees of freedom indices # represent:

$j = 1$: displacement/force in x-direction
$j = 2$: displacement/force in y-direction
$j = 3$: displacement/force in z-direction
$j = 4$: rotation/moment about x-direction
$j = 5$: rotation/moment about y-direction
$j = 6$: rotation/moment about z-direction

Algorithm for problems involving beams

2) Assembly of global matrices:

2.1 Initialization:

$N_{\text{dof}} = 6N$ (total number of degrees of freedom)

$[\mathbf{K}] = [\mathbf{0}]_{N_{\text{dof}} \times N_{\text{dof}}}$

$[\mathbf{M}] = [\mathbf{0}]_{N_{\text{dof}} \times N_{\text{dof}}}$

2.2 Assembly process:

For each element e:

a) Compute rotation matrix:

$\ell = \|\mathbf{X}(\mathbf{T}_{\text{n}}(e,2),:) - \mathbf{X}(\mathbf{T}_{\text{n}}(e,1),:)\|$ (element size ≡ beam length)

$\{\hat{\boldsymbol{\imath}}'\} = \left(\{\mathbf{X}(\mathbf{T}_{\text{n}}(e,2),:)\}^{\text{T}} - \{\mathbf{X}(\mathbf{T}_{\text{n}}(e,1),:)\}^{\text{T}}\right)/\ell$

$\{\hat{\boldsymbol{\jmath}}'\} = \hat{\boldsymbol{\jmath}}'^{\left(\mathbf{T}_{\text{m}}(e)\right)}$

$\{\hat{\boldsymbol{k}}'\} = \{\hat{\boldsymbol{\imath}}'\} \times \{\hat{\boldsymbol{\jmath}}'\}$

$$[\mathbf{R}'] = \begin{bmatrix} \{\hat{\boldsymbol{\imath}}'\} & \{\hat{\boldsymbol{\jmath}}'\} & \{\hat{\boldsymbol{k}}'\} & \cdots & [\mathbf{0}]_{3\times3} & \cdots \\ \cdots & [\mathbf{0}]_{3\times3} & \cdots & \{\hat{\boldsymbol{\imath}}'\} & \{\hat{\boldsymbol{\jmath}}'\} & \{\hat{\boldsymbol{k}}'\} \end{bmatrix}^{\text{T}}$$

$$\mathbf{R}(:,:,e) = \begin{bmatrix} [\mathbf{R}'] & [\mathbf{0}]_{6\times6} \\ [\mathbf{0}]_{6\times6} & [\mathbf{R}'] \end{bmatrix}$$

Algorithm for problems involving beams

b) Compute shape function derivatives:

$$N_{,x'}(1) = -1/\ell$$

$$N_{,x'}(2) = 1/\ell$$

c) Compute each element matrix:

c1) Axial component of stiffness matrix:

$$\mathbf{B}'_{\mathrm{a}}(1,:,e) = [N_{,x'}(1) \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad N_{,x'}(2) \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$\bar{\mathbf{C}}'_{\mathrm{a}} = E^{(\mathrm{T_m}(e))} A^{(\mathrm{T_m}(e))}$$

$$\mathbf{K}_{\mathrm{a}}(:,:,e) = \ell[\mathbf{R}(:,:,e)]^{\mathrm{T}}[\mathbf{B}'_{\mathrm{a}}(1,:,e)]^{\mathrm{T}}[\bar{\mathbf{C}}'_{\mathrm{a}}][\mathbf{B}'_{\mathrm{a}}(1,:,e)][\mathbf{R}(:,:,e)]$$

c2) Bending component of stiffness matrix:

$$\mathbf{B}'_{\mathrm{b}}(:,:,e) = \begin{bmatrix} 0 & 0 & 0 & 0 & N_{,x'}(1) & 0 & 0 & 0 & 0 & 0 & N_{,x'}(2) & 0 \\ 0 & 0 & 0 & 0 & 0 & N_{,x'}(1) & 0 & 0 & 0 & 0 & 0 & N_{,x'}(2) \end{bmatrix}$$

$$\bar{\mathbf{C}}'_{\mathrm{b}} = E^{(\mathrm{T_m}(e))} \begin{bmatrix} I_{y'}^{(\mathrm{T_m}(e))} & 0 \\ 0 & I_{z'}^{(\mathrm{T_m}(e))} \end{bmatrix}$$

$$\mathbf{K}_{\mathrm{b}}(:,:,e) = \ell[\mathbf{R}(:,:,e)]^{\mathrm{T}}[\mathbf{B}'_{\mathrm{b}}(:,:,e)]^{\mathrm{T}}[\bar{\mathbf{C}}'_{\mathrm{b}}][\mathbf{B}'_{\mathrm{b}}(:,:,e)][\mathbf{R}(:,:,e)]$$

Algorithm for problems involving beams

c3) Shear component of stiffness matrix:

$N = 1/2$ (shape functions assuming only 1 Gauss point) **(*)**

$$\mathbf{B}_\mathrm{s}'(:,:,e) = \begin{bmatrix} 0 & N_{,x'}(1) & 0 & 0 & 0 & -N & 0 & N_{,x'}(2) & 0 & 0 & 0 & -N \\ 0 & 0 & N_{,x'}(1) & 0 & N & 0 & 0 & 0 & N_{,x'}(2) & 0 & N & 0 \end{bmatrix}$$

$$\bar{\mathbf{C}}_\mathrm{s}' = G^{(\mathbf{T}_\mathrm{m}(e))} A^{(\mathbf{T}_\mathrm{m}(e))} \begin{bmatrix} k_y^{(\mathbf{T}_\mathrm{m}(e))} & 0 \\ 0 & k_z^{(\mathbf{T}_\mathrm{m}(e))} \end{bmatrix}$$

$$\mathbf{K}_\mathrm{s}(:,:,e) = \ell[\mathbf{R}(:,:,e)]^\mathrm{T}[\mathbf{B}_\mathrm{s}'(:,:,e)]^\mathrm{T}[\bar{\mathbf{C}}_\mathrm{s}'][\mathbf{B}_\mathrm{s}'(:,:,e)][\mathbf{R}(:,:,e)]$$

c4) Torsion component of stiffness matrix:

$$\mathbf{B}_\mathrm{t}'(1,:,e) = [0 \quad 0 \quad 0 \quad N_{,x'}(1) \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad N_{,x'}(2) \quad 0 \quad 0]$$

$$\bar{\mathbf{C}}_\mathrm{t}' = G^{(\mathbf{T}_\mathrm{m}(e))} J^{(\mathbf{T}_\mathrm{m}(e))} k_\mathrm{t}^{(\mathbf{T}_\mathrm{m}(e))}$$

$$\mathbf{K}_\mathrm{t}(:,:,e) = \ell[\mathbf{R}(:,:,e)]^\mathrm{T}[\mathbf{B}_\mathrm{t}'(1,:,e)]^\mathrm{T}[\bar{\mathbf{C}}_\mathrm{t}'][\mathbf{B}_\mathrm{t}'(1,:,e)][\mathbf{R}(:,:,e)]$$

(*) Using only 1 Gauss point yields a sub-integrated element (resulting integral not exact) but avoids the ***shear locking*** problem of the Timoshenko beam model.

Algorithm for problems involving beams

c5) Mass matrix:

$\{\boldsymbol{\xi}\} = \{-1/\sqrt{3}\,;1/\sqrt{3}\}$ (Gauss points coordinates) **(*)**

$\{\boldsymbol{w}\} = \{1;\ 1\}$ (Gauss points weights)

$$\overline{\boldsymbol{\rho}}' = \rho^{(\mathbf{T}_{\mathrm{m}}(e))}\begin{bmatrix} A^{(\mathbf{T}_{\mathrm{m}}(e))} & 0 & 0 & 0 & 0 & 0 \\ 0 & A^{(\mathbf{T}_{\mathrm{m}}(e))} & 0 & 0 & 0 & 0 \\ 0 & 0 & A^{(\mathbf{T}_{\mathrm{m}}(e))} & 0 & 0 & 0 \\ 0 & 0 & 0 & J^{(\mathbf{T}_{\mathrm{m}}(e))} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{y'}^{(\mathbf{T}_{\mathrm{m}}(e))} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{z'}^{(\mathbf{T}_{\mathrm{m}}(e))} \end{bmatrix}$$

$\mathbf{M}_{\mathrm{e}}(:,:,e) = [\mathbf{0}]_{12\times 12}$

For each Gauss point k from 1 to 2

$N(1) = \big(1 - \xi(k)\big)/2$

$N(2) = \big(1 + \xi(k)\big)/2$

$\mathbf{N}(:,:,e,k) = [N(1)[\mathbf{1}]_{6\times 6}\quad N(2)[\mathbf{1}]_{6\times 6}]$ ($[\mathbf{1}]_{6\times 6}$ is the 6×6 identity matrix)

$\mathbf{M}_{\mathrm{e}}(:,:,e) = \mathbf{M}_{\mathrm{e}}(:,:,e) + \boldsymbol{w}(k)\ell[\mathbf{R}(:,:,e)]^{\mathrm{T}}[\mathbf{N}(:,:,e,k)]^{\mathrm{T}}[\overline{\boldsymbol{\rho}}'][\mathbf{N}(:,:,e,k)][\mathbf{R}(:,:,e)]/2$

End loop over Gauss points

(*) 2 Gauss points provide exact integration

Algorithm for problems involving beams

d) Assembly to global matrices:

For each degree of freedom j from 1 to 6

$I_{\mathrm{dof}}(j,1) = 6(\mathbf{T}_{\mathrm{n}}(e,1)-1)+j$

$I_{\mathrm{dof}}(6+j,1) = 6(\mathbf{T}_{\mathrm{n}}(e,2)-1)+j$

End loop over DOFs

$\mathbf{K}(I_{\mathrm{dof}},I_{\mathrm{dof}}) = \mathbf{K}(I_{\mathrm{dof}},I_{\mathrm{dof}}) + \mathbf{K}_{\mathrm{a}}(:,:,e) + \mathbf{K}_{\mathrm{b}}(:,:,e) + \mathbf{K}_{\mathrm{s}}(:,:,e) + \mathbf{K}_{\mathrm{t}}(:,:,e)$

$\mathbf{M}(I_{\mathrm{dof}},I_{\mathrm{dof}}) = \mathbf{M}(I_{\mathrm{dof}},I_{\mathrm{dof}}) + \mathbf{M}_{\mathrm{e}}(:,:,e)$

End loop over elements

Algorithm for problems involving beams

3) Compute global force vector:

3.1 Point loads:

$\{\hat{f}\} = \{\mathbf{0}\}_{N_{\mathrm{dof}}\times 1}$

For row q in $[\mathbf{F}_\mathrm{e}]$

$\hat{\mathbf{f}}(6(\mathbf{F}_\mathrm{e}(q,2)-1)+\mathbf{F}_\mathrm{e}(q,3),1) = \hat{\mathbf{f}}(6(\mathbf{F}_\mathrm{e}(q,2)-1)+\mathbf{F}_\mathrm{e}(q,3),1) + \mathbf{F}_\mathrm{e}(q,1)$

End loop over rows in $[\mathbf{F}_\mathrm{e}]$

3.2 Nodal distributed forces:

$[\mathbf{Q}] = \{\mathbf{0}\}_{N\times 6}$

For row r in $[\mathbf{Q}_\mathrm{e}]$

$\mathbf{Q}\big(\mathbf{Q}_\mathrm{e}(r,2),\mathbf{Q}_\mathrm{e}(r,3)\big) = \mathbf{Q}_\mathrm{e}(r,1)$

End loop over rows in $[\mathbf{Q}_\mathrm{e}]$

3.3 Nodal body forces:

$[\mathbf{B}] = \{\mathbf{0}\}_{N\times 6}$

For row s in $[\mathbf{B}_\mathrm{e}]$

$\mathbf{B}\big(\mathbf{B}_\mathrm{e}(s,2),\mathbf{B}_\mathrm{e}(s,3)\big) = \mathbf{B}_\mathrm{e}(s,1)$

End loop over rows in $[\mathbf{B}_\mathrm{e}]$

Algorithm for problems involving beams

3.4 Assembly process:

For each element e:

a) Compute element force vector:

$\mathbf{b}(:,e) = \{\mathbf{B}(\mathbf{T}_\mathrm{n}(e,1),:),\mathbf{B}(\mathbf{T}_\mathrm{n}(e,2),:)\}^\mathrm{T}$

$\mathbf{q}(:,e) = \{\mathbf{Q}(\mathbf{T}_\mathrm{n}(e,1),:),\mathbf{Q}(\mathbf{T}_\mathrm{n}(e,2),:)\}^\mathrm{T}$

$\hat{\mathbf{f}}_\mathrm{e}(:,e) = [\mathbf{M}_\mathrm{e}(:,:,e)]\{\mathbf{b}(:,e)\}$

For each Gauss point k (from 1 to 2):

$\hat{\mathbf{f}}_\mathrm{e}(:,e) = \hat{\mathbf{f}}_\mathrm{e}(:,e) + \boldsymbol{w}(k)\ell[\mathbf{R}(:,:,e)]^\mathrm{T}[\mathbf{N}(:,:,e,k)]^\mathrm{T}[\mathbf{N}(:,:,e,k)][\mathbf{R}(:,:,e)]\{\mathbf{q}(:,e)\}/2$

End loop over Gauss points

b) Assembly to global force vector:

For each degree of freedom j from 1 to 6

$I_\mathrm{dof}(j,1) = 6(\mathbf{T}_\mathrm{n}(e,1)-1)+j$

$I_\mathrm{dof}(6+j,1) = 6(\mathbf{T}_\mathrm{n}(e,2)-1)+j$

End loop over DOFs

$\hat{\mathbf{f}}(I_\mathrm{dof},1) = \hat{\mathbf{f}}(I_\mathrm{dof},1) + \hat{\mathbf{f}}_\mathrm{e}(:,e)$

End loop over elements

Algorithm for problems involving beams

4) Boundary conditions:

4.1 Initialization:

$\{\hat{\boldsymbol{u}}\} = \{\mathbf{0}\}_{N_{\mathrm{dof}}\times 1}$

4.2 Prescribed and free DOFs:

For row p in $[\mathbf{U}_{\mathrm{p}}]$

$\boldsymbol{I}_{\mathrm{p}}(p) = 6\big(\mathbf{U}_{\mathrm{p}}(p,2) - 1\big) + \mathbf{U}_{\mathrm{p}}(p,3)$ (vector with prescribed degrees of freedom)

$\hat{\boldsymbol{u}}\big(\boldsymbol{I}_{\mathrm{p}}(p),1\big) = \mathbf{U}_{\mathrm{p}}(p,1)$

End loop over rows in $[\mathbf{U}_{\mathrm{p}}]$

$\boldsymbol{I}_{\mathrm{f}} = \{1{:}\,N_{\mathrm{dof}}\} - \{\boldsymbol{I}_{\mathrm{p}}\}$ (**Tip**: in Matlab, this operation can be done with the **`setdiff`** function)

```
If = setdiff(1:Ndof,Ip)
```

5) Solve system of equations (static case):

5.1 Solve system:

$\hat{\boldsymbol{u}}(\boldsymbol{I}_{\mathrm{f}},1) = [\mathbf{K}(\boldsymbol{I}_{\mathrm{f}},\boldsymbol{I}_{\mathrm{f}})]^{-1}\big(\hat{\boldsymbol{f}}(\boldsymbol{I}_{\mathrm{f}},1) - [\mathbf{K}(\boldsymbol{I}_{\mathrm{f}},\boldsymbol{I}_{\mathrm{p}})]\{\hat{\boldsymbol{u}}(\boldsymbol{I}_{\mathrm{p}},1)\}\big)$ (displacements/rotations at free DOFs)

$\hat{\boldsymbol{f}}_{\mathrm{R}} = [\mathbf{K}]\{\hat{\boldsymbol{u}}\} + \{\hat{\boldsymbol{f}}\}$ (reaction forces/moments at prescribed DOFs)

Algorithm for problems involving beams

4) Boundary conditions:

4.1 Initialization:

$\{\hat{\boldsymbol{u}}\} = \{\mathbf{0}\}_{N_{\mathrm{dof}}\times 1}$

4.2 Prescribed and free DOFs:

For row p in $[\mathbf{U}_{\mathrm{p}}]$

$\boldsymbol{I}_{\mathrm{p}}(p) = 6\big(\mathbf{U}_{\mathrm{p}}(p,2) - 1\big) + \mathbf{U}_{\mathrm{p}}(p,3)$ (vector with prescribed degrees of freedom)

$\hat{\boldsymbol{u}}\big(\boldsymbol{I}_{\mathrm{p}}(p),1\big) = \mathbf{U}_{\mathrm{p}}(p,1)$

End loop over rows in $[\mathbf{U}_{\mathrm{p}}]$

$\boldsymbol{I}_{\mathrm{f}} = \{1{:}\,N_{\mathrm{dof}}\} - \{\boldsymbol{I}_{\mathrm{p}}\}$ (**Tip**: in Matlab, this operation can be done with the **`setdiff`** function)

```
If = setdiff(1:Ndof,Ip)
```

5) Solve system of equations (static case):

5.1 Solve system:

$\hat{\boldsymbol{u}}(\boldsymbol{I}_{\mathrm{f}},1) = [\mathbf{K}(\boldsymbol{I}_{\mathrm{f}},\boldsymbol{I}_{\mathrm{f}})]^{-1}\big(\hat{\boldsymbol{f}}(\boldsymbol{I}_{\mathrm{f}},1) - \big[\mathbf{K}\big(\boldsymbol{I}_{\mathrm{f}},\boldsymbol{I}_{\mathrm{p}}\big)\big]\big\{\hat{\boldsymbol{u}}\big(\boldsymbol{I}_{\mathrm{p}},1\big)\big\}\big)$ (displacements/rotations at free DOFs)

$\hat{\boldsymbol{f}}_{\mathrm{R}} = [\mathbf{K}]\{\hat{\boldsymbol{u}}\} + \{\hat{\boldsymbol{f}}\}$ (reaction forces/moments at prescribed DOFs)

Algorithm for problems involving beams

6) Postprocess: Computing strain and internal forces in beam elements

6.1 Get strain at each beam element:

For each element e:

a) Get element displacements:

For each degree of freedom j from 1 to 6

$$I_{\mathrm{dof}}(j,1) = 6(\mathbf{T}_{\mathrm{n}}(e,1)-1)+j$$
$$I_{\mathrm{dof}}(6+j,1) = 6(\mathbf{T}_{\mathrm{n}}(e,2)-1)+j$$

End loop over DOFs

$$\{\hat{\boldsymbol{u}}_e\} = \{\hat{\boldsymbol{u}}(I_{\mathrm{dof}},1)\}$$

b) Get each strain component:

$$\boldsymbol{\varepsilon}'_{\mathrm{a}}(1,e) = [\mathbf{B}'_{\mathrm{a}}(1,:,e)][\mathbf{R}(:,:,e)]\{\hat{\boldsymbol{u}}_e\}$$
$$\boldsymbol{\varepsilon}'_{\mathrm{s}}(:,e) = [\mathbf{B}'_{\mathrm{s}}(:,:,e)][\mathbf{R}(:,:,e)]\{\hat{\boldsymbol{u}}_e\}$$
$$\bar{\boldsymbol{\varepsilon}}'_{\mathrm{t}}(1,e) = [\mathbf{B}'_{\mathrm{t}}(1,:,e)][\mathbf{R}(:,:,e)]\{\hat{\boldsymbol{u}}_e\}$$
$$\bar{\boldsymbol{\varepsilon}}'_{\mathrm{b}}(:,e) = [\mathbf{B}'_{\mathrm{b}}(:,:,e)][\mathbf{R}(:,:,e)]\{\hat{\boldsymbol{u}}_e\}$$

The stress and strain at a given point y', z' on the section can be obtained as:

$$\varepsilon_{x'x'}(y',z',e) = \boldsymbol{\varepsilon}'_{\mathrm{a}}(1,e) + z'\bar{\boldsymbol{\varepsilon}}'_{\mathrm{b}}(1,e) - y'\bar{\boldsymbol{\varepsilon}}'_{\mathrm{b}}(2,e)$$
$$\gamma_{x'y'}(y',z',e) = \boldsymbol{\varepsilon}'_{\mathrm{s}}(1,e) - z'\bar{\boldsymbol{\varepsilon}}'_{\mathrm{t}}(1,e)$$
$$\gamma_{x'z'}(y',z',e) = \boldsymbol{\varepsilon}'_{\mathrm{s}}(2,e) + y'\bar{\boldsymbol{\varepsilon}}'_{\mathrm{t}}(1,e)$$

$$\sigma_{x'x'}(y',z',e) = E^{(\mathbf{T}_{\mathrm{m}}(e))}\varepsilon_{x'x'}(y',z',e)$$
$$\tau_{x'y'}(y',z',e) = G^{(\mathbf{T}_{\mathrm{m}}(e))}\gamma_{x'y'}(y',z',e)$$
$$\tau_{x'z'}(y',z',e) = G^{(\mathbf{T}_{\mathrm{m}}(e))}\gamma_{x'z'}(y',z',e)$$

Algorithm for problems involving beams

c) Get internal forces and moments at each element node:

$$\hat{\boldsymbol{f}}'_{\mathrm{int}}(:,e) = [\mathbf{R}(:,:,e)][\mathbf{K}_{\mathrm{a}}(:,:,e) + \mathbf{K}_{\mathrm{b}}(:,:,e) + \mathbf{K}_{\mathrm{s}}(:,:,e) + \mathbf{K}_{\mathrm{t}}(:,:,e)]\{\hat{\boldsymbol{u}}_e\}$$

$$F_{x'}(:,e) = \left\{\hat{\boldsymbol{f}}'_{\mathrm{int}}(1,e); -\hat{\boldsymbol{f}}'_{\mathrm{int}}(7,e)\right\}$$

$$F_{y'}(:,e) = \left\{\hat{\boldsymbol{f}}'_{\mathrm{int}}(2,e); -\hat{\boldsymbol{f}}'_{\mathrm{int}}(8,e)\right\}$$

$$F_{z'}(:,e) = \left\{\hat{\boldsymbol{f}}'_{\mathrm{int}}(3,e); -\hat{\boldsymbol{f}}'_{\mathrm{int}}(9,e)\right\}$$

$$M_{x'}(:,e) = \left\{\hat{\boldsymbol{f}}'_{\mathrm{int}}(4,e); -\hat{\boldsymbol{f}}'_{\mathrm{int}}(10,e)\right\}$$

$$M_{y'}(:,e) = \left\{\hat{\boldsymbol{f}}'_{\mathrm{int}}(5,e); -\hat{\boldsymbol{f}}'_{\mathrm{int}}(11,e)\right\}$$

$$M_{z'}(:,e) = \left\{\hat{\boldsymbol{f}}'_{\mathrm{int}}(6,e); -\hat{\boldsymbol{f}}'_{\mathrm{int}}(12,e)\right\}$$

End loop over elements