# CIVIL 449: Nonlinear Analysis of Structures

School of Architecture, Civil & Environmental Engineering
Civil Engineering Institute

## Distributed plasticity – Fiber-based elements
## Element formulations and sectional analysis

Prof. Dr. Dimitrios Lignos & Dr. Diego Heredia Rosa
EPFL, ENAC, IIC, RESSLab

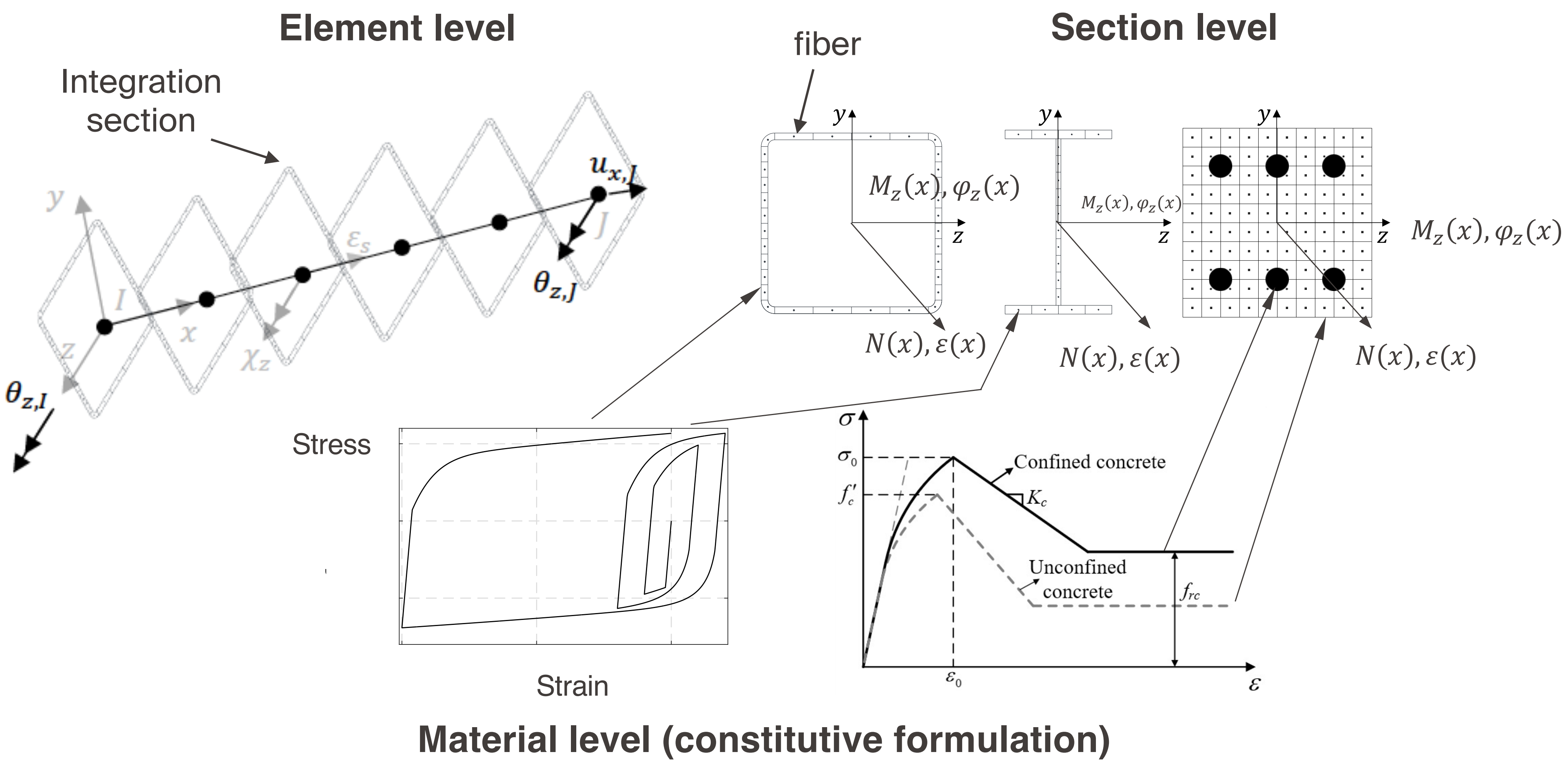
# Objectives of today's lecture

To introduce:

- Fiber-based beam-column elements
- Fiber discretization of cross sections
- Constitutive models for fiber-based elements
- Computation of input strains
- Section analysis
- Type of element formulations
  - Displacement-based beam-column elements
  - Force-based beam-column elements
- Integration methods for member forces and member stiffness

This week's material

# Fiber-based beam-column elements: basic workflow

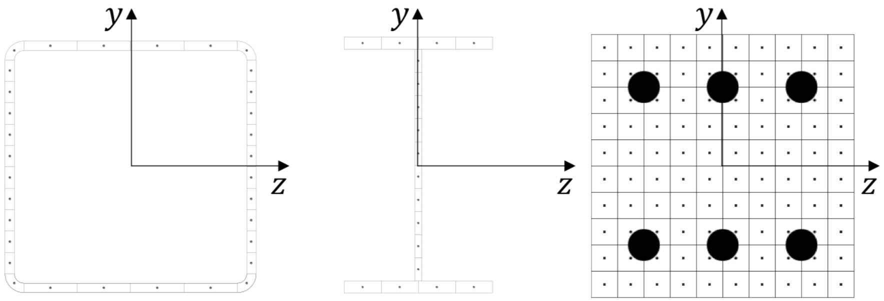


# State determination of fiber section

- The section deformations at the corresponding integration point of an element,

$$\mathbf{d}_s(x) = \{\varepsilon(x) \quad \varphi_z(x)\}^T$$

- A section is meshed into smaller blocks called fibers.




- The strain of each fiber is determined by section deformation vector $\mathbf{d}_s(x)$ and its coordinates $(y_{fiber}, z_{fiber})$ along the section.
- The strain of the *k*-th fiber in a cross section,

$$\varepsilon_{k.fiber} = \varepsilon(x) + \varphi_z(x) y_{k.fiber} = \{1, y_{k.fiber}\} \cdot \begin{Bmatrix} \varepsilon(x) \\ \varphi_z(x) \end{Bmatrix} \quad \text{(2-d element)}$$

$$\varepsilon_{k.fiber} = \varepsilon(x) + \varphi_z(x) y_{k.fiber} + \varphi_y(x) z_{k.fiber} = \{1, y_{k.fiber}, -z_{k.fiber}\} \cdot \begin{Bmatrix} \varepsilon(x) \\ \varphi_z(x) \\ \varphi_y(x) \end{Bmatrix} \quad \text{(3-d element)}$$

# State determination of a fiber section (2)

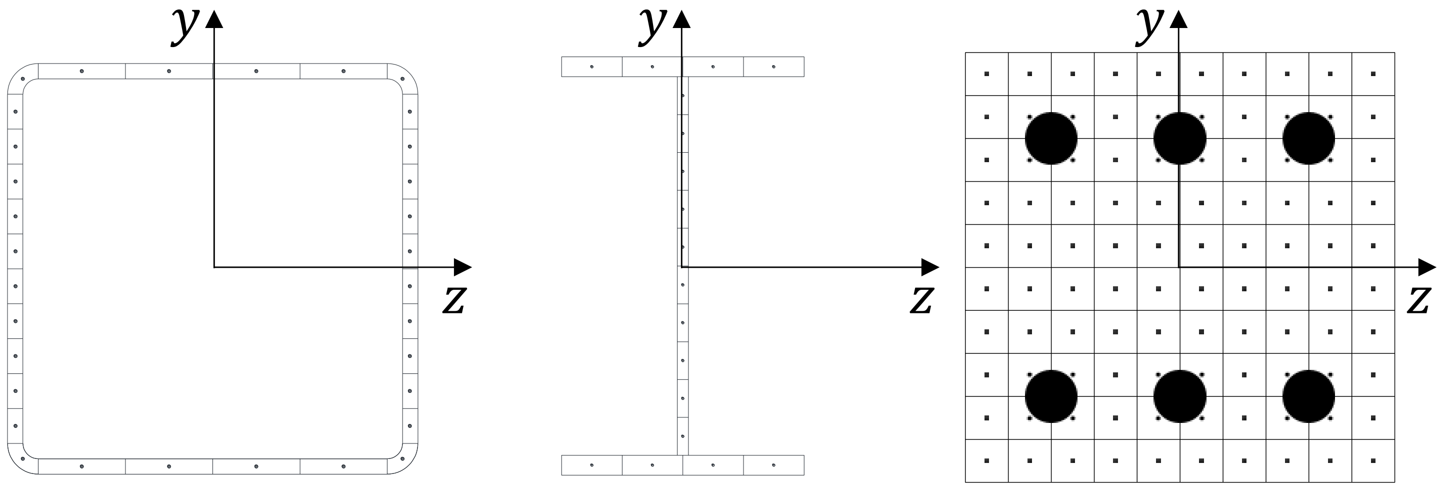
- Section geometrical vector (careful with the local coordinate system of your cross-section discretization)

2-D beam-column element

$$\mathbf{l}_{k.fiber} = \{1, y_{k.fiber}\}$$

3-D beam-column element

$$\mathbf{l}_{k.fiber} = \{1, y_{k.fiber}, -z_{k.fiber}\}$$



# State determination of a fiber section (2)

- For a 2d element, the section resisting force vector $\mathbf{Q}_s$ is defined as

$$\mathbf{Q}_s = [N(x), M_z(x)]^T$$

- Similarly, the section stiffness matrix $\mathbf{k}_s$ is defined as

$$\mathbf{k}_s = \begin{bmatrix} \mathrm{EA} & \mathrm{EQ_z} \\ \mathrm{EQ_z} & \mathrm{EI_z} \end{bmatrix}$$

# State determination of a fiber section (3)

- Using the sign convention presented in the previous slide, the following structural mechanics equations are recalled:

$$I_z = \int_A y^2 dA \,, I_y = \int_A z^2 dA \,, Q_z = \int_A y dA \,, Q_y = \int_A z dA$$

$$N = \int_A \sigma dA \,, M_z = \int_A y \sigma dA \,, M_y = \int_A -z \sigma dA$$

- These can be computed numerically using the values from each fiber

$$I_z = \int_A y^2 dA = \sum_{ifib=1}^{nfib} y_{ifib}^2 \cdot A_{ifib} \,, \qquad Q_z = \int_A y dA = \sum_{ifib=1}^{nfib} y_{ifib} \cdot A_{ifib}$$

$$N = \int_A \sigma dA = \sum_{ifib=1}^{nfib} \sigma_{ifib} \cdot A_{ifib} \,, \qquad M_z = \int_A y \sigma dA = \sum_{ifib=1}^{nfib} y_{ifib} \cdot \sigma_{ifib} \cdot A_{ifib}$$

# State determination of a fiber section (4)

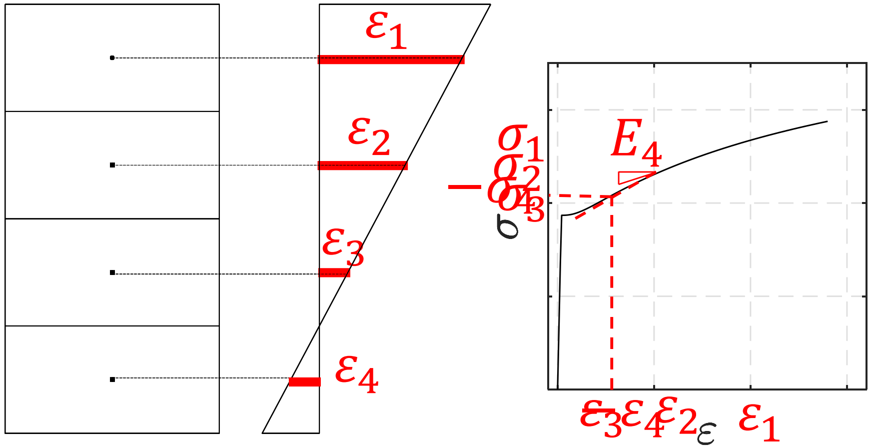
- The section resisting force vector and stiffness matrix can therefore be rewritten as

$$\mathbf{Q}_s = \begin{bmatrix} N \\ M_z \end{bmatrix} = \begin{bmatrix} \sum_{ifib=1}^{nfib} \sigma_{ifib} \cdot A_{ifib} \\ \sum_{ifib=1}^{nfib} y_{ifib} \cdot \sigma_{ifib} \cdot A_{ifib} \end{bmatrix} = \sum_{k=1}^{nfib} \mathbf{l}_{ifib}^T \cdot (\sigma_{ifib} A_{ifib})$$

$$\mathbf{k}_s = \begin{bmatrix} \mathrm{EA} & \mathrm{EQ_z} \\ \mathrm{EQ_z} & \mathrm{EI_z} \end{bmatrix} = \begin{bmatrix} \mathrm{E} \sum_{ifib=1}^{nfib} A_{ifib} & \mathrm{E} \sum_{ifib=1}^{nfib} y_{ifib} \cdot A_{ifib} \\ \mathrm{E} \sum_{ifib=1}^{nfib} y_{ifib} \cdot A_{ifib} & \mathrm{E} \sum_{ifib=1}^{nfib} y_{ifib}^2 \cdot A_{ifib} \end{bmatrix} = \sum_{k=1}^{nfib} \mathbf{l}_{ifib}^T \cdot (E_{ifib} A_{ifib}) \cdot \mathbf{l}_{ifib}$$

# State determination of fiber section (5)

- Based on the material constitutive formulation and the fiber strain, the tangent modulus $E_{k.fiber}$ and stress, $\sigma_{k.fiber}$ of the *k*-th fiber can be determined.
- $E_{k.fiber}$ and $\sigma_{k.fiber}$ are used to compute the section stiffness and forces.

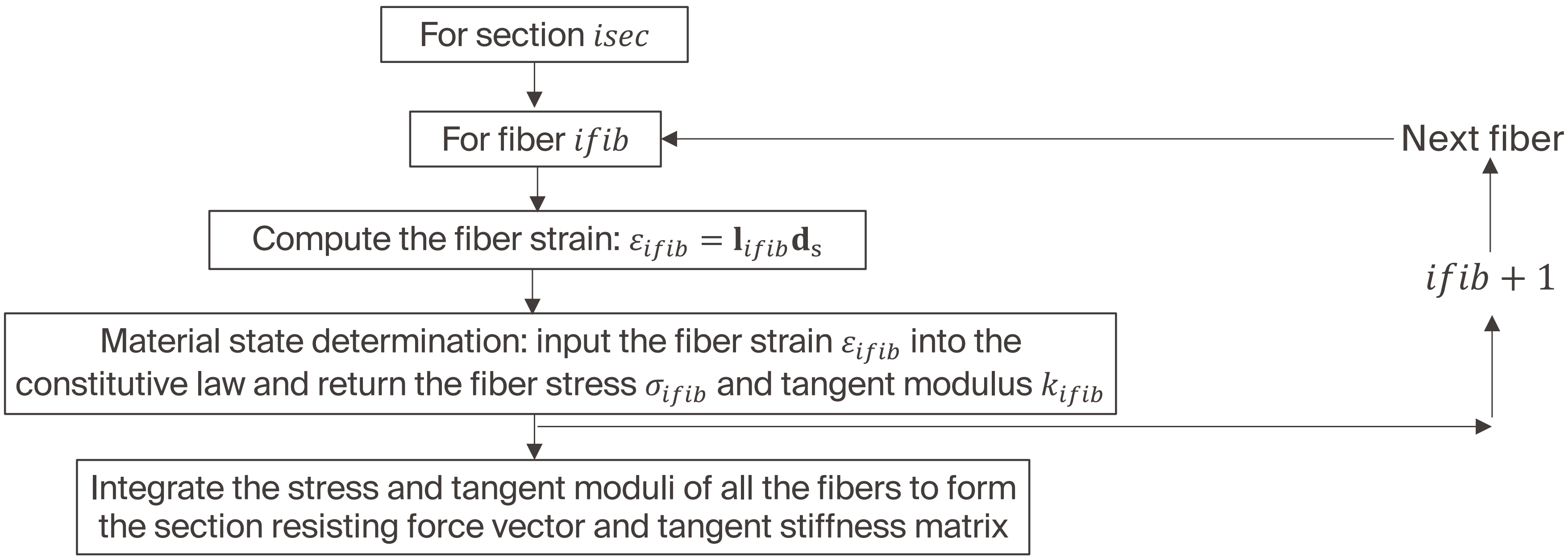



Section resisting force vector: $\mathbf{Q_s}(x) = \mathbf{l}_1^T \cdot (\sigma_1 A_1) + \mathbf{l}_2^T \cdot (\sigma_2 A_2) + \mathbf{l}_3^T \cdot (\sigma_3 A_3) + \mathbf{l}_4^T \cdot (\sigma_4 A_4)$

Section stiffness matrix: $\mathbf{k_s}(x) = \mathbf{l}_1^T \cdot (E_1 A_1) \cdot \mathbf{l}_1 + \mathbf{l}_2^T \cdot (E_2 A_2) \cdot \mathbf{l}_2 + \mathbf{l}_3^T \cdot (E_3 A_3) \cdot \mathbf{l}_3 + \mathbf{l}_4^T \cdot (E_4 A_4) \cdot \mathbf{l}_4$

# State determination of a fiber section (6)

- The flow chart gives the solution process for the section state determination of a fiber section



# Beam-column element types for input strain field computation

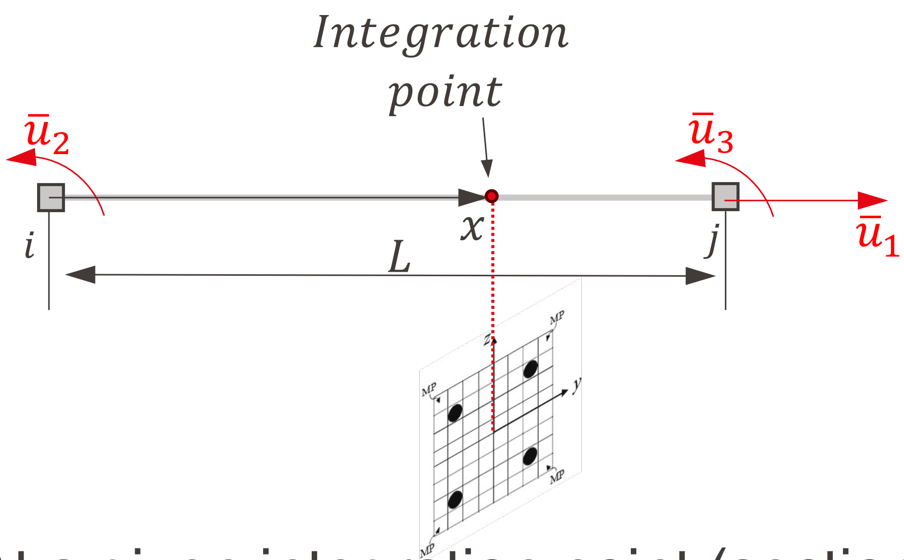
- **If element deformations are the primary unknowns, then the element is called displacement-based (stiffness-based)**
  - Uses cubic interpolation functions to approximate the displacement field
  - Assumed interpolation functions for displacement field result in linear curvature (i.e., constant axial strain along the element length)
  - Requires many element segments to capture the strain gradient along a member
- **If the element forces are the primary unknowns, then the element is called force-based (flexibility-based)**
  - Uses linear interpolation functions to approximate the force field
  - Usually requires a single element with few sections (i.e., integration points) to represent a member
  - Keeps the number of degrees of freedom to a minimum
  - Usually, the element is analyzed without rigid body modes (basic reference frame)
  - State determination is challenging because (a) the flexibility matrix and (b) the deformation vector that corresponds to the applied forces should be computed at every solution step.

# EPFL Beam-column element types for input strain field computation (2)

- Due to the element formulation, the element state determination for **force-based elements** necessities an iterative solution scheme to determine the element resisting forces and stiffness matrix. Typically, a Newton-Raphson algorithm is used. This iterative scheme will be denoted by the index $j$
- Conversely, the element state determination for **displacement-based elements** does not use an iterative solution
- Recall from the previous lectures that the element state determination procedure is performed at each iteration $i$ of the load-displacement control algorithm at every load or displacement increment $n$

# Displacement-based beam-column element

- In the basic reference frame, the vectors of nodal displacements, $\bar{\mathbf{u}}$, and element resisting forces $\bar{\mathbf{q}}$, are as follows:




$$\bar{\mathbf{u}} = \{\bar{u}_1, \bar{u}_2, \bar{u}_3\}^T$$

$$\bar{\mathbf{q}} = \{\bar{q}_1, \bar{q}_2, \bar{q}_3\}^T$$

- At a given integration point (section):

$$d_a(x) = N_1(x)\bar{u}_1$$

$$d_f(x) = N_2(x)\bar{u}_2 + N_3(x)\bar{u}_3$$

# Displacement-based beam-column element (2)

- The vector of displacements anywhere inside the element can be expressed as follows:

$$\bar{\mathbf{d}}(x) = \underbrace{\begin{bmatrix} N_1(x) & 0 & 0 \\ 0 & N_2(x) & N_3(x) \end{bmatrix}}_{\bar{\mathbf{a}}_d\,(x)} \cdot \bar{\mathbf{u}}$$

$\bar{\mathbf{a}}_d(x)$ : Matrix containing the linear interpolation functions for the axial displacement and the cubic interpolation functions for the rotations

# Displacement-based beam-column element (3)

- The interpolation functions for the uniaxial bending case (from week #3):

axial

$$N_1(x) = \frac{x}{L}$$

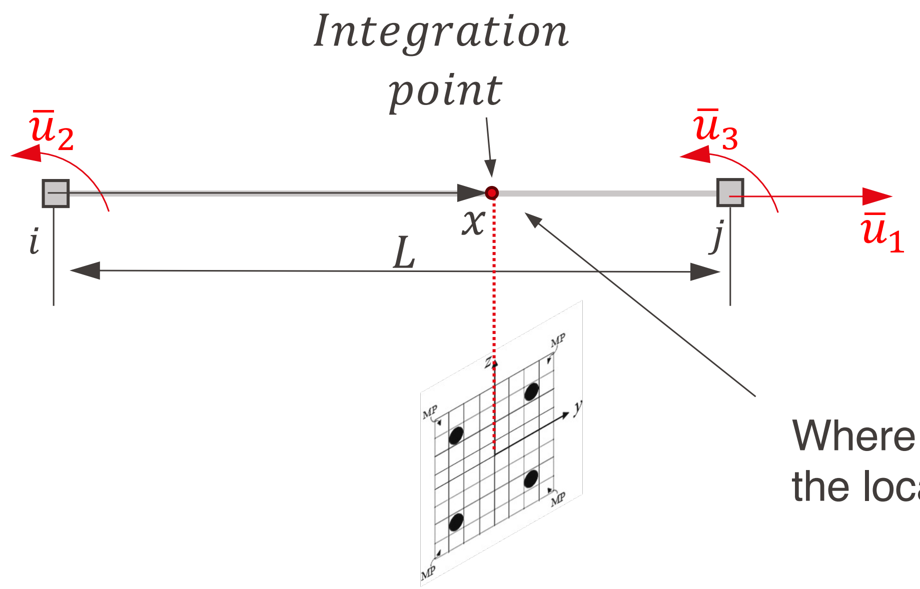
rotational

$$N_2(x) = \frac{x^3}{L^2} - \frac{2x^2}{L} + x$$

$$N_3(x) = \frac{x^3}{L^2} - \frac{x^2}{L}$$

# Displacement-based beam-column element (4)

- The element is divided into segments according to selected integration points.
- The behavior at each integration point is characterized by the section deformation vector $\mathbf{d}_s(x)$ and the section resisting force $\mathbf{Q}_{sr}(x)$ :




$$\mathbf{d}_s(x) = \{\varepsilon(x), \varphi_y(x)\}^T$$

$$\mathbf{Q}_{sr}(x) = \{N(x), M_y(x)\}^T$$

Where *x* is the coordinate of a section under consideration in the local coordinate system

# Displacement-based beam-column element (5)

- The section deformations $\mathbf{d}_s(x)$ can be written as follows:

$$\mathbf{d}_s(x) = \{\varepsilon(x), \varphi_y(x)\}^T = \left\{\frac{\vartheta d_a(x)}{\vartheta x} \quad \frac{\vartheta^2 d_f(x)}{\vartheta x^2}\right\}^T$$

- And in a matrix form by using the shape functions:

$$\mathbf{d}_s(x) = \underbrace{\begin{bmatrix} N_1'(x) & 0 & 0 \\ 0 & N_2''(x) & N_3''(x) \end{bmatrix}}_{\overline{\mathbf{B}}(x)} \cdot \overline{\mathbf{u}}$$

# Displacement-based beam-column element (6)

- If plane sections remain plane,

$$\mathbf{d}_s^{n,i}(x) = \overline{\mathbf{B}}(x) \cdot \overline{\mathbf{u}}^{n,i}$$

- Where $\overline{\mathbf{B}}(x)$ is the matrix derived from the derivatives of displacement interpolation functions.

$$\overline{\mathbf{B}}(x) = \begin{bmatrix} N_1'(x) & 0 & 0 \\ 0 & N_2''(x) & N_3''(x) \end{bmatrix}$$

# State determination of displacement-based element

- The tangent element stiffness matrix at iteration $i$ of step $n$, $\overline{\mathbf{K}}^{n,i}$, of a displacement-based beam-column element of length $L$, and the element resisting force vector $\overline{\mathbf{q}}^{n,i}$ can be expressed as follows:

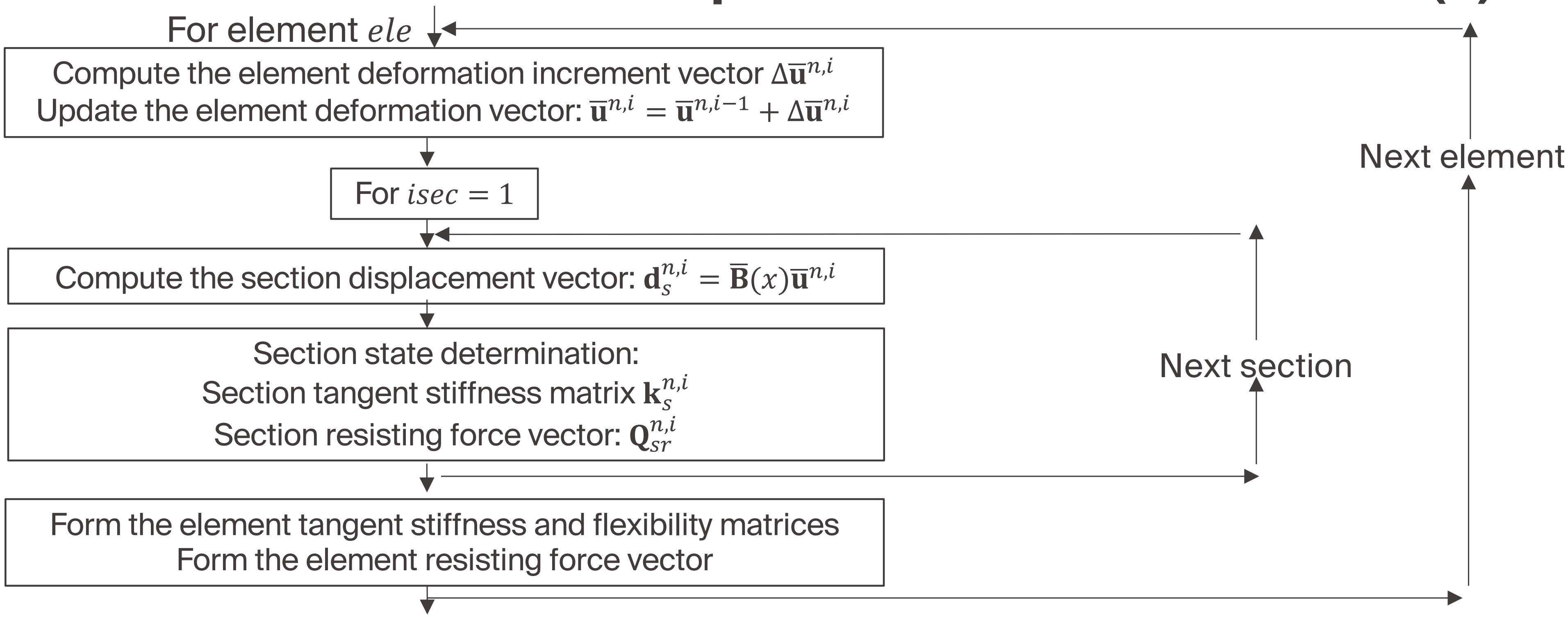
$$\overline{\mathbf{K}}^{n,i} = \int_0^L \overline{\mathbf{B}}^T(x) \cdot \mathbf{k}_s^{n,i}(x) \cdot \overline{\mathbf{B}}(x) \cdot dx$$

$$\overline{\mathbf{q}}^{n,i} = \int_0^L \overline{\mathbf{B}}^T(x) \cdot \mathbf{Q}_{\mathrm{sr}}^{n,i}(x) \cdot dx$$

We calculate those numerically with some numerical integration schemes

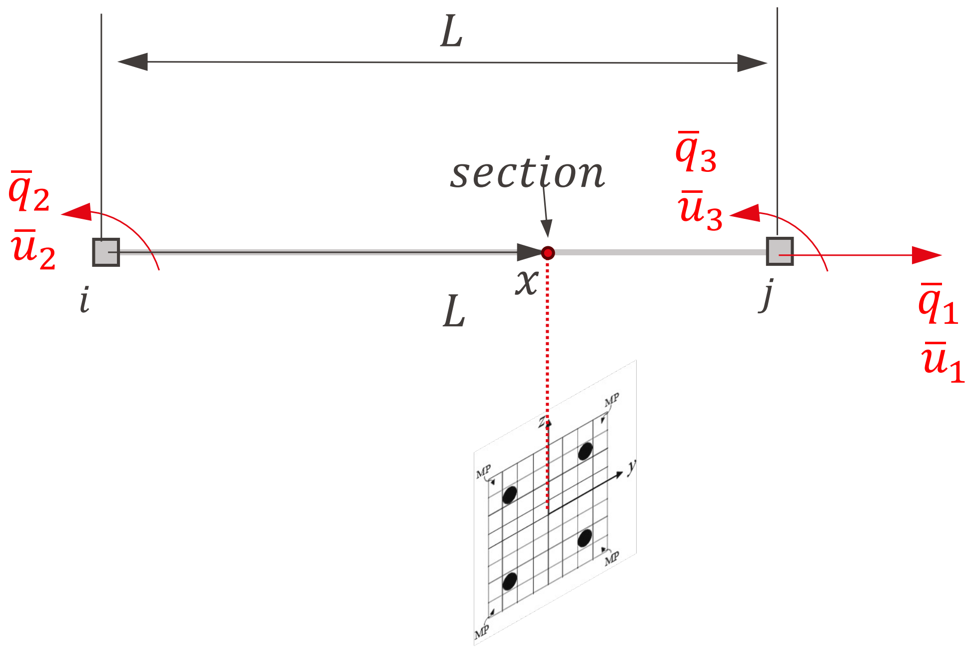
- Gauss-Legendre,
- Gauss-Lobatto,
- Gauss Radau,
- midpoint rule

# State determination of displacement-based element (2)



# Force-based (or flexibility-based) elements

- The element vector generalized nodal forces $\overline{\mathbf{q}}$ at the basic reference frame (without rigid body modes) is as follows:




$$\overline{\mathbf{q}} = \{\bar{q}_1, \bar{q}_2, \bar{q}_3\}^T$$

$$\overline{\mathbf{u}} = \{\bar{u}_1, \bar{u}_2, \bar{u}_3\}^T$$

# Force-based elements (2)

- We assume that the bending moment distribution inside the element is linear and that the axial force distribution is constant. In vector notation, the internal forces become,

$$\mathbf{Q}_s^{n,i,j}(x) = \mathbf{b}(x) \cdot \overline{\mathbf{q}}^{n,i,j}$$

(internal forces) (nodal forces)

- $\mathbf{b}(x)$ is a matrix containing the force interpolation functions (see next slide) relating the generalized nodal forces $\overline{\mathbf{q}}$ to the internal forces $\mathbf{Q}_s(x)$.

# Force-based elements (3)

- The vector of force interpolation functions is as follows for the case of no element loads (concentrated and/or distributed):

$$\mathbf{b}(x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \left(\frac{x}{L} - 1\right) & \left(\frac{x}{L}\right) \end{bmatrix}$$

# Force-based elements (4)

- The section flexibility matrix $\mathbf{f}_s^{n,i,j}(x) = \left(\mathbf{k}_s^{n,i,j}\right)^{-1}(x)$ , such that,

$$\mathbf{d}_s^{n,i,j}(x) = \mathbf{f}_s^{n,i,j}(x) \cdot \mathbf{Q}_s^{n,i,j}(x)$$

(section deformations) (internal forces)

- The incremental version of the previous equation is as follows,

$$\Delta\mathbf{d}_s^{n,i,j}(x) = \mathbf{f}_s^{n,i,j-1}(x) \cdot \Delta\mathbf{Q}_s^{n,i,j}(x) = \mathbf{f}_s^{n,i,j-1}(x) \cdot \mathbf{b}(x) \cdot \Delta\overline{\mathbf{q}}^{n,i,j}(x)$$

(incremental section deformations) (incremental element forces)

- This formulation satisfies the element equilibrium in a strict sense even if the element softens when deformed beyond its ultimate resistance.

# Force-based elements (5)

- The field of element deformation increment at step $i$ is:

$$\mathbf{\Delta d}_s^{n,i,j}(x) = \mathbf{f}_s^{n,i,j-1}(x)\mathbf{\Delta Q}_s^{n,i,j}(x)$$

- Therefore, the deformation update at step $i$ in incremental form becomes

$$\mathbf{d}_s^{n,i,j}(x) = \mathbf{d}_s^{n,i,j-1}(x) + \mathbf{\Delta d}_s^{n,i,j}(x)$$

# State determination of force-based element

- The section flexibility at iteration $j$, $\mathbf{f}_s^{n,i,j}(x)$ is,

$$\mathbf{f}_s^{n,i,j}(x) = \left(\mathbf{k}_s^{n,i,j}(x)\right)^{-1}$$

- Element flexibility matrix, $\overline{\boldsymbol{F}}^{n,i,j}$ at iteration $j$ is:

$$\overline{\boldsymbol{F}}^{n,i,j} = \int_0^L \mathbf{b}^T(x) \cdot \mathbf{f}_s^{n,i,j}(x) \cdot \mathbf{b}(x) dx$$

- The element stiffness matrix, $\overline{\boldsymbol{K}}^{n,i,j}$ at iteration $j$ is:

$$\overline{\boldsymbol{K}}^{n,i,j} = \left(\mathbf{F}^{n,i,j}\right)^{-1}$$

- The element end displacements at iteration $j$ is,

$$\overline{\mathbf{u}}^{n,i,j} = \int_0^L \mathbf{b}^T(x) \cdot \mathbf{d}_s^{n,i,j}(x)\, dx$$

(end displacements) (section deformations)

# State determination of force-based element (2)

- The linearization of the equation in Slide 23 yields to,

$$\overline{\mathbf{F}}^{n,i,j} \cdot \Delta\overline{\mathbf{q}}^{n,i,j} = \overline{\mathbf{u}}_{\mathrm{u}}^{n,i,j}$$

(Flexibility matrix) (Residual displacements)

- Where $\Delta\overline{\mathbf{q}}^{n,i,j}$ and $\overline{\mathbf{u}}_{\mathrm{u}}^{n,i,j}$ are vectors of the force increments and residual displacements, respectively.

- A meaningful expression for the flexibility matrix $\overline{\mathbf{F}}^{n,i,j}$ can only be derived for the beam-column element in the basic element system.

# State determination of force-based element (3)

- With a known element displacement increment, $\boldsymbol{\Delta\bar{u}}^{n,i,j}$ for the current iteration $j$, the element force-displacement relation becomes,

$$\boldsymbol{\Delta\bar{q}}^{n,i,j} = \left(\bar{\mathbf{F}}^{n,i,j-1}\right)^{-1} \cdot \boldsymbol{\Delta\bar{u}}^{n,i,j}$$

- The internal force increment at iteration $j$ can then be computed

$$\boldsymbol{\Delta Q}_s^{n,i,j}(x) = \mathbf{b}(x)\boldsymbol{\Delta\bar{q}}^{n,i,j}$$

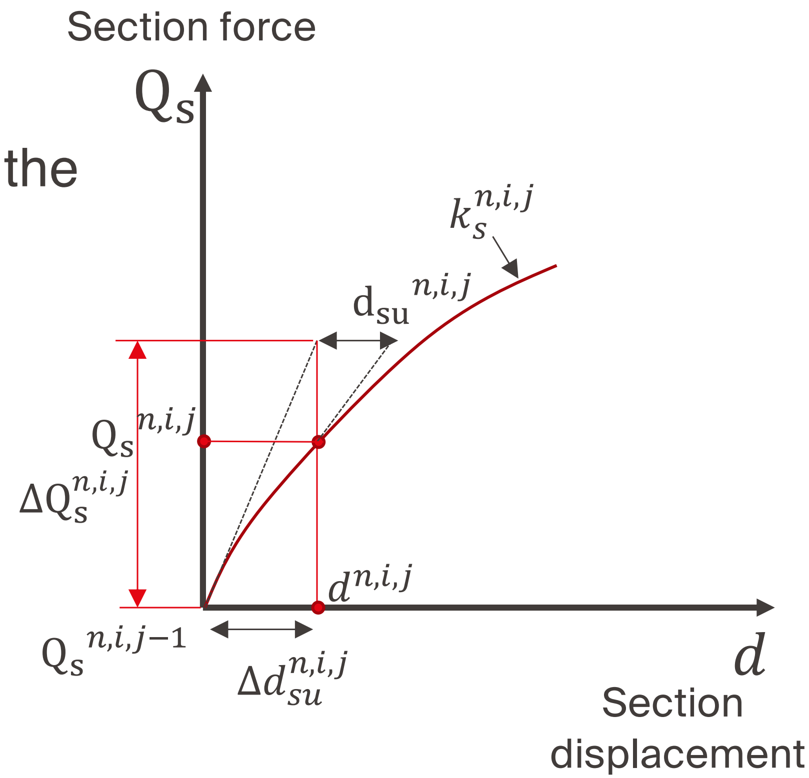
# State determination of force-based element (4)

- The residual section deformations are computed as follows:

$$\mathbf{d}_{su}^{n,i,j}(x) = \mathbf{f}_s(x)^{n,i,j}[\mathbf{Q}_s^{n,i,j}(x) - \mathbf{Q}_{sr}^{n,i,j}(x)]$$

- Therefore, based on the principle of virtual forces, the residual element displacements at iteration $j$ are:

$$\overline{\mathbf{u}}_u^{n,i,j} = \int_0^L \mathbf{b}^T(x)\mathbf{d}_{su}^{n,i,j}(x)\,dx$$



# State determination of force-based element (5)

- These residuals are transformed to residual forces such that the element resisting forces become,

$$\overline{\mathbf{q}}^{n,i,j} = \overline{\mathbf{q}}^{n,i,j-1} - \left(\overline{\mathbf{F}}^{n,i,j}\right)^{-1} \overline{\mathbf{u}}_{u}^{n,i,j}$$

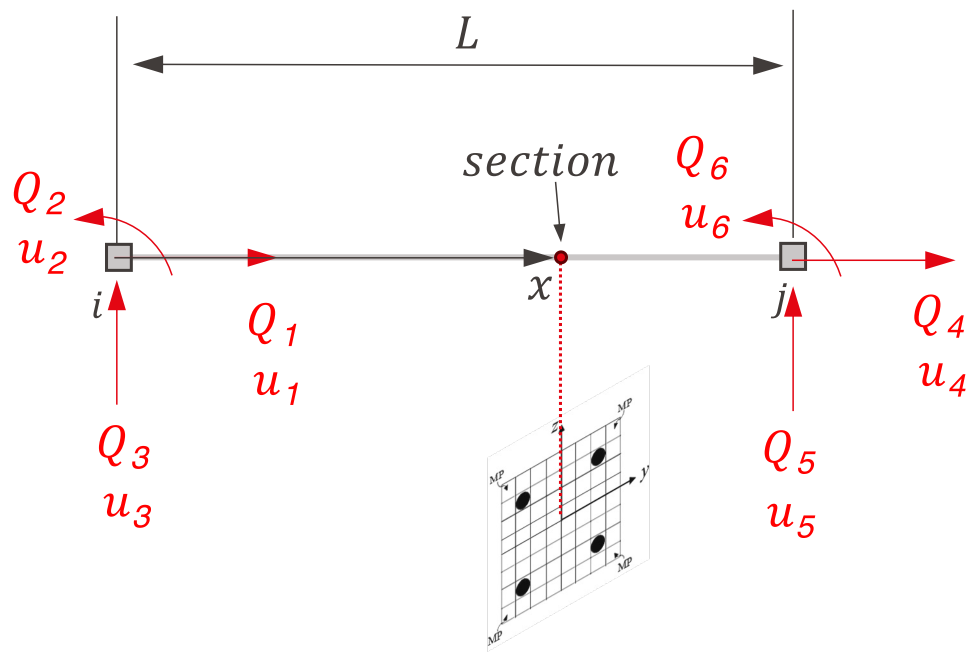
- The unbalanced section forces are as follows:

$$\mathbf{Q}_{\mathrm{su}}^{n,i,j}(x) = \mathbf{Q}_{s}^{n,i,j}(x) - \mathbf{Q}_{\mathrm{sr}}^{n,i,j}(x)$$

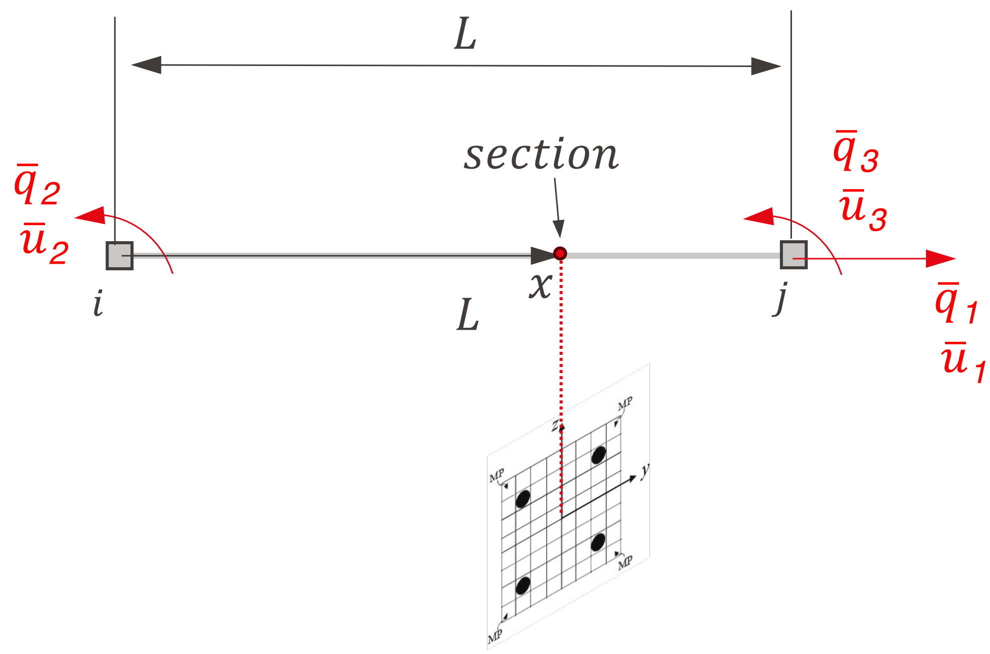
- Continue accordingly to reduce the unbalanced section forces with Newton-Raphson by using a tolerance

# State determination of force-based element (6)

Local reference frame
Beam element with rigid body modes




Basic reference frame
Beam element without rigid body modes



# State determination of force-based element (8)

For element $ele$

Compute the element deformation increment vector $\Delta\bar{\mathbf{u}}^{n,i}$
Update the element deformation vector: $\bar{\mathbf{u}}^{n,i} = \bar{\mathbf{u}}^{n,i-1} + \Delta\bar{\mathbf{u}}^{n,i}$

Set $j = 1$

Compute the element force increment vector $\Delta\bar{\mathbf{q}}^{n,i,j} = \bar{\boldsymbol{K}}^{n,i,j}\Delta\bar{\mathbf{u}}^{n,i,j}$
Update the element force vector: $\bar{\mathbf{q}}^{n,i,j} = \bar{\mathbf{q}}^{n,i,j-1} + \Delta\bar{\mathbf{q}}^{n,i,j}$

For $isec = 1$

Compute the section force increment vector $\Delta\boldsymbol{Q}_s^{n,i,j} = \boldsymbol{b}(x)\Delta\bar{\mathbf{q}}^{n,i,j}$
Compute the section displacement increment vector $\Delta\boldsymbol{d}_s^{n,i,j} = \boldsymbol{d}_{su}^{n,i,j-1} + \mathbf{f}_s^{n,i,j-1}\Delta\boldsymbol{Q}_s^{n,i,j}$
Update the section force vector: $\boldsymbol{Q}_s^{n,i,j} = \boldsymbol{Q}_s^{n,i,j-1} + \Delta\boldsymbol{Q}_s^{n,i,j}$
Update the section displacement vector: $\boldsymbol{d}_s^{n,i,j} = \boldsymbol{d}_s^{n,i,j-1} + \Delta\boldsymbol{d}_s^{n,i,j}$

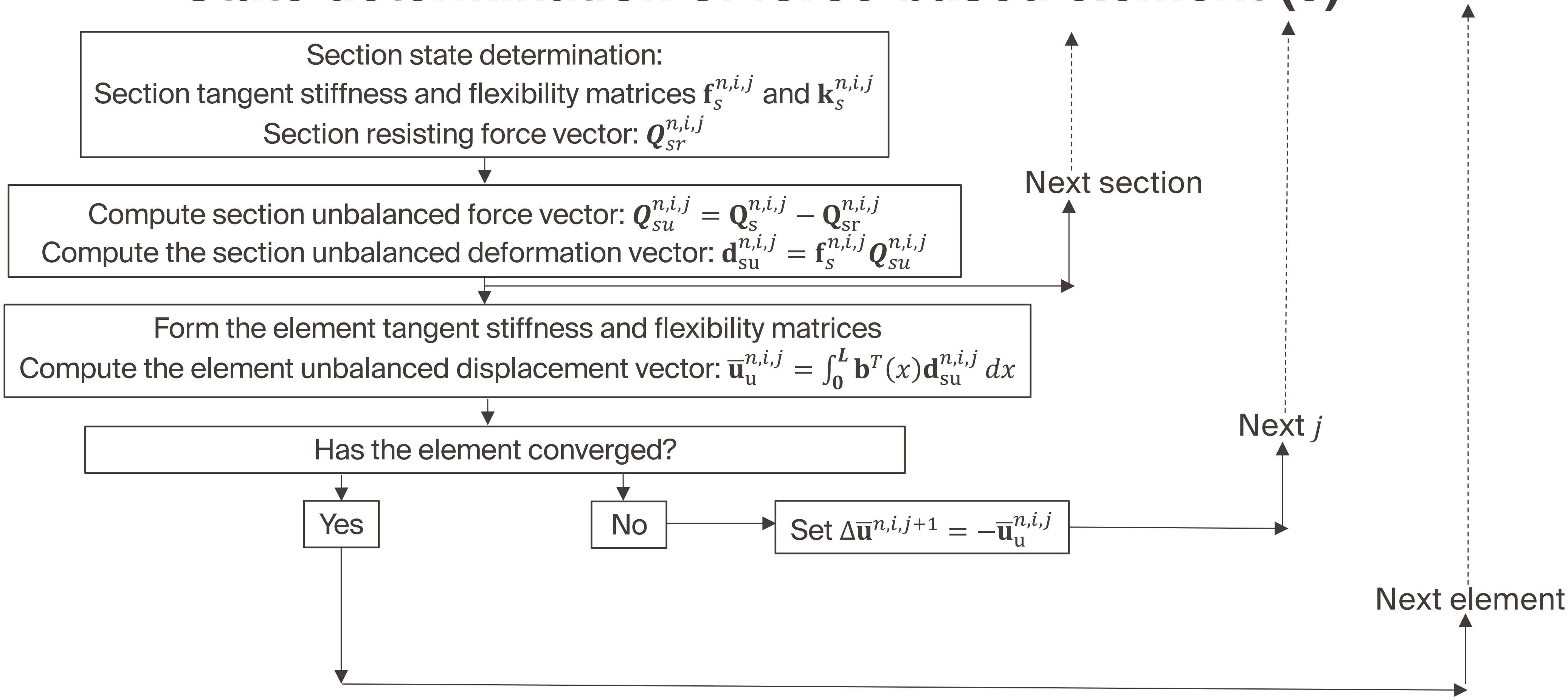
Section state determination

Next section

Next $j$

Next element

# State determination of force-based element (9)



# Application of element loads : Force-based elements

- For **force-based** beam-column elements, the following steps are taken to apply element loads:
- The section forces are computed using

$$\boldsymbol{Q}_s^{n,i,j} = \mathbf{b}(x)\Delta\overline{\mathbf{q}}^{n,i,j} + \mathbf{b}_g(x)\boldsymbol{W}^n$$

$\mathbf{b}_g$ is a force transformation matrix that relates the applied element loads $\boldsymbol{W}^n$ to the element forces in a beam without rigid body modes

- Once the element state determination has converged, the element resisting force vector in the local reference frame $\boldsymbol{Q}^{n,i}$ is computed using

$$\boldsymbol{Q}^{n,i} = \mathbf{L}\overline{\mathbf{q}}^{n,i,j} + \mathbf{t}_g(x)\boldsymbol{W}^n$$

$\mathbf{t}_g$ is a transformation matrix that depends on the applied element loads $\boldsymbol{W}^n$

# Application of element loads : Force-based elements (2)

- For uniformly distributed loads $W^n = \left[w_x^n, w_y^n\right]^T$, the matrices $\boldsymbol{b}_g$ and $\boldsymbol{t}_g$ follow the assumed sign convention and are given as follows:

$$\mathbf{b}_g(x) = \begin{bmatrix} L - x & 0 \\ 0 & \frac{L}{2}x(L - x) \end{bmatrix}$$

$$\mathbf{t}_g(x) = \begin{bmatrix} -L & 0 & 0 & 0 & 0 & 0 \\ 0 & -L/2 & 0 & 0 & -L/2 & 0 \end{bmatrix}$$

- For different load cases, these matrices can be derived from equilibrium considerations

# Application of element loads : Displacement-based elements

- For **displacement-based** beam-column elements, element loads should be transformed into nodal loads and applied accordingly
- This may require a finer discretization into elements

# Example: OpenSees

## EleLoad Command

Command_Manual https://opensees.berkeley.edu/wiki/index.php/EleLoad_Command

The eleLoad command is used to construct an ElementalLoad object and add it to the enclosing LoadPattern.

```
load $eleLoad $arg1 $arg2 $arg3 ....
```

The element loads are only applied to line elements. Continuum elements do not accept element loads. When NDM=2, the beam column elements all accept eleLoad commands of the following form:

```
eleLoad -ele $eleTag1 <$eleTag2 ....> -type -beamUniform $Wy <$Wx>
```

```
eleLoad -range $eleTag1 $eleTag2 -type -beamPoint $Py $xL <$Px>
```

When NDM=3, the beam column elements all accept eleLoad commands of the following form:

```
eleLoad -ele $eleTag1 <$eleTag2 ....> -type -beamUniform $Wy $Wz <$Wx>
```

```
eleLoad -range $eleTag1 $eleTag2 -type -beamPoint $Py $Pz $xL <$Px>
```

| | |
|---|---|
| **$eleTag1 $eleTag2 ...** | tag of PREVIOUSLY DEFINED element |
| **$Wx** | mag of uniformily distributed ref load acting in direction along member length |
| **$Wy** | mag of uniformily distributed ref load acting in local y direction of element |
| **$Wz** | mag of uniformily distributed ref load acting in local z direction of element |
| **$Py** | mag of ref point load acting in direction along member length |
| **$Py** | mag of ref point load acting in local y direction of element |
| **$Pz** | mag of ref point load acting in local z direction of element |
| **$xL** | location of point load relative to node I, prescribed as fraction of element length |