Elementi di Meccanica Computazionale
Corso di Laurea in Ingegneria Civile
Pavia, 2014

Linear beams: direct approach and finite element formulation

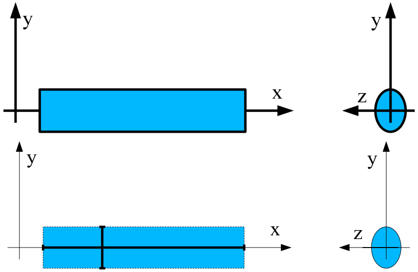
Ferdinando Auricchio [1] [2]

[1]DICAR – Dipartimento di Ingegneria Civile e Architettura, Università di Pavia, Italy
[2]IMATI – Istituto di Matematica Applicata e Tecnologie Informatiche, CNR, Italy

March 13, 2014

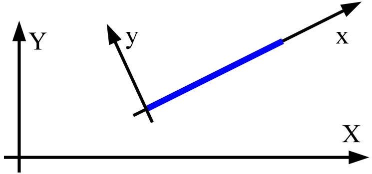
Beam theory: geometry I

- **Beam geometric definition:**
  - 3D body with two dimensions negligeble wrt the third one (1D problem)
  - Distinguish between beam **axis** (x-axis) and beam **cross-section** (yz-plane)
  - Consider only planar beam problems
    - assume symmetry wrt xy-plane (load and cross section)
    - y-z are principal axes of inertia
    - develop all the problem only considering x-y axes



Beam theory: local vs global I

- Axes $x - y$ are local beam axes, i.e. relative to a specific beam under investigation
- Local beam axes $x - y$ distinguished from global axes $X - Y$



- Global axes are relative to the whole problem and not to a specific beam
  - Developments relative to beam model are done initially wrt local beam axes, later on converted to global axes

Change of coordinates I

- Given two bases $\{\mathbf{e}_1, \mathbf{e}_2\}$ and $\{\mathbf{E}_1, \mathbf{E}_2\}$, a vector $\mathbf{f}$ can be expressed equivalently in any of the two bases as

$$\mathbf{f} = F_b \mathbf{E}_b = f_a \mathbf{e}_a$$

  - Adopt Einstein summation convention, i.e. repeated index in a multiplicative term implies summation over valid index range

- Possible to derive a relationship between the two set of components, by taking dot product with one of the base vectors

$$(F_b \mathbf{E}_b) \cdot \mathbf{E}_i = (f_a \mathbf{e}_a) \cdot \mathbf{E}_i$$

resulting in

$$F_i = f_a (\mathbf{e}_a \cdot \mathbf{E}_i)$$

or

$$\boxed{F_i = R_{ia} f_a \quad \text{with} \quad R_{ia} = \mathbf{E}_i \cdot \mathbf{e}_a} \tag{1}$$

Change of coordinates II

- Introducing engineering (vector) notation, $\mathbf{f}$ is expressed in the two bases as two sets of numbers, i.e. $\{\mathbf{f}\}$ and $\{\mathbf{F}\}$ with

$$\{\mathbf{f}\}|_i = f_i \quad , \quad \{\mathbf{F}\}|_i = F_i$$

- Possible to convert components also through the following matrix operation

$$\boxed{\{\mathbf{F}\} = [\mathbf{R}]\{\mathbf{f}\}} \qquad (2)$$

where

$$[\mathbf{R}] = \begin{bmatrix} \mathbf{E}_1 \cdot \mathbf{e}_1 & \mathbf{E}_1 \cdot \mathbf{e}_2 & \mathbf{E}_1 \cdot \mathbf{e}_3 \\ \mathbf{E}_2 \cdot \mathbf{e}_1 & \mathbf{E}_2 \cdot \mathbf{e}_2 & \mathbf{E}_2 \cdot \mathbf{e}_3 \\ \mathbf{E}_3 \cdot \mathbf{e}_1 & \mathbf{E}_3 \cdot \mathbf{e}_2 & \mathbf{E}_3 \cdot \mathbf{e}_3 \end{bmatrix}$$

☐ **Exercise.** Comment on the differences between the quantities $\mathbf{f}$, $\{\mathbf{f}\}$ and $\{\mathbf{F}\}$.

☐ **Exercise.** Show that in the case of rotation around the third axis, $[\mathbf{R}]$ can also be rappresented as follows:

$$[\mathbf{R}] = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

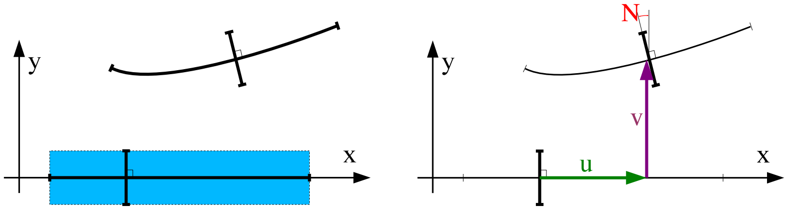
with ϕ the angle between $\mathbf{E}_1$ and $\mathbf{e}_1$.

☐ **Exercise.** Prove that $\mathbf{R}$ is an orthogonal matrix, i.e. $[\mathbf{R}]^T[\mathbf{R}] = [\mathbf{R}][\mathbf{R}]^T = [\mathbf{I}]$

Beam theory: kinematics I

- **Kinematic assumptions:** [Euler-Bernoulli beam]

Plane sections normal to beam axis remain plane and normal to axis during change of configuration



- **Continuum displacement field:** (seen as a 2D continuum problem)

$$\mathbf{s} = \mathbf{s}(x, y) \quad \text{with} \quad \mathbf{s} = \left\{ \begin{array}{c} s_x(x, y) \\ s_y(x, y) \end{array} \right\}$$

- **Beam “axis” displacements:** (seen as a 1D beam problem)

$$u = u(x) \ , \ v = v(x) \ , \ \theta = \theta(x)$$

Beam theory: kinematics I

- Express continuum field $\mathbf{s} = \{s_x, s_y\}^T$ in terms of beam axis displacements u, v, θ

$$\begin{cases} s_x(x,y) = u(x) - y\theta(x) \\ s_y(x,y) = v(x) \end{cases} \tag{3}$$

  with
  - displacement positive if directed as coordinate axes
  - rotation positive if counterclockwise

- Unknown fields:

$$u = u(x) \quad , \quad v = v(x) \quad , \quad \theta = \theta(x)$$

★ $\mathbf{s} = \mathbf{s}(x,y)$ in term of $u = u(x)$, $v = v(x)$, $\theta = \theta(x)$ has a linear dependence on y

- So far, we are requiring that plane sections remain plane but we are not requiring that sections remain orthogonal to beam axis
- For notational simplicity apices indicate derivation wrt independent variable, i.e.

$$u' = \frac{du}{dx}$$

Beam: kinematics I

- **Strain:**
  - Non trivial strain components: ε_{xx} & ε_{yx}

$$\begin{cases} \varepsilon_{xx} = \dfrac{ds_x}{dx} = u' - y\theta' \\ \varepsilon_{yx} = \dfrac{1}{2}\left(\dfrac{ds_y}{dx} + \dfrac{ds_x}{dy}\right) = \dfrac{1}{2}\left(v' - \theta\right) \end{cases}$$

  - Introduce beam strain-like quantities:

$$\begin{cases} \varepsilon = u' & \text{axial strain} \\ \chi = \theta' & \text{curvature} \\ \gamma = v' - \theta & \text{shear strain} \end{cases}$$

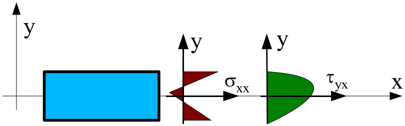
$$\boxed{\begin{cases} \varepsilon_{xx} = \varepsilon - y\chi \\ \varepsilon_{yx} = \dfrac{1}{2}\gamma \end{cases}}$$

☐ **Exercise.** Verify that other strain components besides ε_{xx} and ε_{yx} are effectively zero

Beam: static relations I

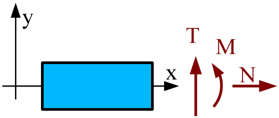
- **Stress:**
  - Assume only two non-trivial stress components

$$\sigma_{xx} \neq 0 \quad , \quad \tau_{yx} \neq 0$$



  - As a consequence, only three non-trivial "internal forces" or "characteristics" (force-like quantities):

$$\begin{cases} N = \int_A [\sigma_{xx}] \, dA & \text{axial force} \\ M = -\int_A [y\sigma_{xx}] \, dA & \text{bending moment} \\ S = \int_A [\tau_{yx}] \, dA & \text{shear force} \end{cases}$$



Beam: equilibrium I

- **Note:** we are assuming as positive beam characteristics the standard ones !!
- **Equilibrium equations:**

$$\begin{cases} N' + p = 0 \\ M' - S = 0 \\ S' - q = 0 \end{cases} \tag{4}$$

with p and q axial and transverse distributed loads (positive if along axis directions)

- ☐ **Exercise.** Recall definition of stress tensor components in a general framework
- ☐ **Exercise.** Recall definition of (internal forces) characteristic components in a general framework
- ☐ **Exercise.** Verify equations 4 imposing directly equilibrium of an infinitesimal beam element
- ☐ **Exercise.** Verify equations 4 starting from a three-dimensional version of principle of virtual work using assumed expression for displacement fields and integrating by parts. Observe that not only it is possible to verify equilibrium equations but this approach leads naturally to a definition of internal characteristic forces "conjugate" to assumed displacements

Beam: constitutive relation I

- **Pointwise 3D constitutive** stress-strain relations:

$$\begin{cases} \sigma_{xx} = E\varepsilon_{xx} = E\left(\varepsilon - y\chi\right) \\ \sigma_{yx} = 2G\varepsilon_{yx} = G\gamma \end{cases} \tag{5}$$

  - Using beam characteristic definition, obtain **local beam constitutive** relations:

$$\begin{cases} N = \int_A \left[\sigma_{xx}\right] dA & \Rightarrow & N = EA\varepsilon \\ M = -\int_A \left[y\sigma_{xx}\right] dA & \Rightarrow & M = EI\chi \\ S = \int_A \left[\sigma_{yx}\right] dA & \not\Rightarrow & S = \kappa GA\gamma \end{cases}$$

with

$$\begin{cases} I = \int_A y^2 dA & \text{moment of inertia} \\ \kappa & \text{shear correction factor} \end{cases}$$

    - **Note:** shear correction factor κ introduced to correct inconsistency associated to equation 5 (for rectangular cross sections $\kappa = 5/6$)

Beam: basic equations I

Kinematics:

$$\begin{cases} s_x = u(x) - y\theta(x) \\ s_y = v(x) \end{cases}$$

$$\begin{cases} \varepsilon = u' \\ \chi = \theta' \\ \gamma = v' - \theta \end{cases}$$

Constitutive:

$$\begin{cases} N = EA\varepsilon \\ M = EI\chi \\ S = \kappa GA\gamma \end{cases}$$

Equilibrium:

$$\begin{cases} N' + p = 0 \\ M' - S = 0 \\ S' - q = 0 \end{cases}$$

☐ **Exercise.** Review 3D Saint Venant model. Compare developed model with 3D Saint Venant model.

Beam: differential eqns I

- **Bernoulli beam:** beam with no shear deformation

Plane sections normal to beam axis remain plane and normal to axis during the deformation

$$\gamma = 0 \quad \Rightarrow \quad \theta = v'$$

- **Differential eqns** (Elastica):

$$\left.\begin{array}{l} M'' - q = 0 \\ M = EIv'' \end{array}\right\} \quad \Rightarrow \quad \boxed{EIv^{IV} - q = 0}$$

☐ **Exercise.** Using the elastica compute rotations at the extremes of a simply supported beam loaded with a unit moment on one support

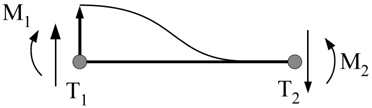
☐ **Exercise.** Using the elastica compute the maximum deflection of a simply supported beam loaded in the middle span with a unit concentrated force.

Beam: direct stiffness formulation I

- Using elastica equation, we can investigate the stiffness of a given beam element (*displacement based approach*)
- **Stiffness:** (set of) force(s) required to obtain a unitary displacement
- Indicating with pedix 1 quantities relative to left node (node 1) and with pedix 2 quantities relative to right node (node 2), we can start to solve problems such as:

$$v_1 = 1 \quad , \quad \theta_1 = v_2 = \theta_2 = 0$$

- Using the elastica, compute beam characteristic forces T_1, M_1, T_2, M_2



$$T_1 = 12\frac{EI}{l^3} \qquad M_1 = -6\frac{EI}{l^2}$$
$$T_2 = 12\frac{EI}{l^3} \qquad M_2 = 6\frac{EI}{l^2} \qquad (6)$$

Beam: direct stiffness formulation I

- **Exercise.** Verify positions 6 using Matlab.

```
%
% Computation of stiffness coeff. for a doubly-clamped beam
% subjected to a vertical unit displ. on the left extreme
%

clear

syms x                         % indipendent variable
syms EI                        % cross-section stiffness constant
syms c0 c1 c2 c3               % integration constants
syms l                         % beam length
syms v1 theta1 v2 theta2       % nodal dofs

v = c0 + c1*x + c2*x^2 + c3*x^3

rot  = diff(v  ,x)             % rotation
chi  = diff(rot,x)             % curvature
mom  = EI*chi                  % bending moment
shear= diff(mom,x)             % shear

% impose boundary conditions

eq1=subs(v  ,x,0)-v1
eq2=subs(rot,x,0)-theta1
eq3=subs(v  ,x,l)-v2
eq4=subs(rot,x,l)-theta2
```

Beam: direct stiffness formulation II

```
[c0,c1,c2,c3] = solve(eq1,eq2,eq3,eq4,c0,c1,c2,c3)

v = c0 + c1*x + c2*x^2 + c3*x^3

v = subs(v,{v1,theta1,v2,theta2},[1,0,0,0])

rot   = diff(v  ,x)          % rotation
chi   = diff(rot,x)          % curvature
mom   = EI*chi               % bending moment
shear= diff(mom,x)           % shear

subs(mom  ,x,0)
subs(shear,x,0)
subs(mom  ,x,l)
subs(shear,x,l)
```

Beam: direct stiffness formulation III

- **Exercise.** Verify positions 6 using Sage.

EU beam stiffness

```
#
# Computation of stiffness coeff. for a doubly-clamped beam
# subjected to a vertical unit displ. on the left extreme
#
```

```
var('x')                    # indipendent variable
var('EI')                   # cross-section stiffness constant
cc=var('c1 c2 c3 c4')       # integration constants
var('l')                    # beam length
v(x,cc)=c1*x^3+c2*x^2+c3*x+c4
```

```
rot  = diff(v,x)              # rotation
chi = diff(rot,x)             # curvature
mom  = EI*chi                 # bending moment
shear= diff(mom,x)            # shear
```

```
v;rot;mom;shear
```

```
    (x, cc) |--> c1*x^3 + c2*x^2 + c3*x + c4
    (x, cc) |--> 3*c1*x^2 + 2*c2*x + c3
    (x, cc) |--> 2*(3*c1*x + c2)*EI
    (x, cc) |--> 6*EI*c1
```

```
eq1 = v(0)==1
eq2 = rot(0) == 0
eq3 = v(l)== 0
eq4 = rot(l) ==0
```

```
solution=solve([eq1,eq2,eq3,eq4],c1,c2,c3,c4,solution_dict =
False)
```

```
solution[0]
```

```
    [c1 == 2/l^3, c2 == -3/l^2, c3 == 0, c4 == 1]
```

```
type(solution)
```

```
    <class 'sage.structure.sequence.Sequence'>
```

```
solution[0][0]; solution[0][0].rhs()
```

```
    c1 == 2/l^3
    2/l^3
```

```
c1 = solution[0][0].rhs()
c2 = solution[0][1].rhs()
c3 = solution[0][2].rhs()
c4 = solution[0][3].rhs()
```

```
v1 = function('v1',x)
rot1 = function('rot1',x)
mom1 = function('mom1',x)
shear1 = function('shear1',x)
```

```
v1(x)=c1*x^3+c2*x^2+c3*x+c4
```

```
rot1  = diff(v1,x)            # rotation
chi1  = diff(rot1,x)          # curvature
mom1  = EI*chi1               # bending moment
shear1= diff(mom1,x)          # shear
```

```
v1
```

```
    x |--> -3*x^2/l^2 + 2*x^3/l^3 + 1
```

```
shear1(0);mom1(0);shear1(l);mom1(l)
```

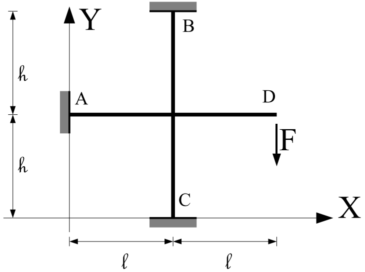
```
    12*EI/l^3
    -6*EI/l^2
    12*EI/l^3
    6*EI/l^2
```

Beam: direct stiffness formulation IV

- **Exercise.** Study the following three cases:

$$v_2 = 1 \ , \ \theta_1 = v_1 = \theta_2 = 0$$
$$\theta_1 = 1 \ , \ v_1 = v_2 = \theta_2 = 0$$
$$\theta_2 = 1 \ , \ v_1 = v_2 = \theta_1 = 0$$

- **Exercise.** Using a displacement based approach solve the following structure and in particular compute the vertical displacement and the rotation of cross section D.



- **Exercise.** Comment on the possible relation between the method adopted to solve the previous exercise and the calculation of the reaction forces for imposed displacement such as $v_1 = 1 \ , \ \theta_1 = v_2 = \theta_2 = 0$

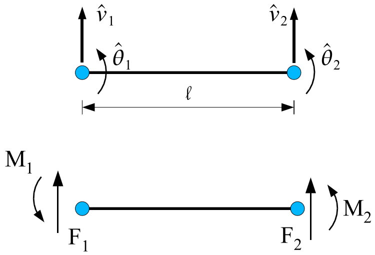
Beam: direct stiffness formulation I

- Using elastica equation, we can compute the complete (!) **beam element stiffness**
- Indicating general "positive" set of beam nodal degrees of freedom as

$$\hat{\mathbf{v}} = \left\{ \hat{v}_1, \hat{\theta}_1, \hat{v}_2, \hat{\theta}_2 \right\}^T$$

and corresponding nodal reactions as

$$\mathbf{F} = \{F_1, M_1, F_2, M_2\}^T$$



we can try to construct relation between $\hat{\mathbf{v}}$ and $\mathbf{F}$

Beam: direct stiffness formulation I

- Relation between $\hat{\mathbf{v}}$ and $\mathbf{F}$ is clearly linear:

$$\boxed{\mathbf{F} = \mathbf{K}^e \hat{\mathbf{v}}}$$

with $\mathbf{K}^e$ indicating the **beam stiffness matrix**

- **Note:** we are now considering positive reaction forces wrt local coordinate system
- **Note:** difference between positive reaction forces and positive beam characteristics
- Accordingly, from equation 6 we can compute the first line of the beam stiffness matrix $\mathbf{K}^e$:

$$\mathbf{K}^e = \begin{bmatrix} 12\frac{EI}{l^3} & 6\frac{EI}{l^2} & -12\frac{EI}{l^3} & 6\frac{EI}{l^2} \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix}$$

Beam: direct stiffness formulation I

☐ **Exercise.** Studying the following three cases:

$$v_2 = 1 \text{ and } \theta_1 = v_1 = \theta_2 = 0$$
$$\theta_1 = 1 \text{ and } v_1 = v_2 = \theta_2 = 0$$
$$\theta_2 = 1 \text{ and } v_1 = v_2 = \theta_1 = 0$$

verify that the beam stiffness matrix $\mathbf{K}^e$ has the following form:

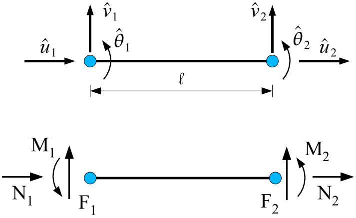
$$\mathbf{K}^e = \begin{bmatrix} 12\frac{EI}{l^3} & 6\frac{EI}{l^2} & -12\frac{EI}{l^3} & 6\frac{EI}{l^2} \\ 6\frac{EI}{l^2} & 4\frac{EI}{l} & -6\frac{EI}{l^2} & 2\frac{EI}{l} \\ -12\frac{EI}{l^3} & -6\frac{EI}{l^2} & 12\frac{EI}{l^3} & -6\frac{EI}{l^2} \\ 6\frac{EI}{l^2} & 2\frac{EI}{l} & -6\frac{EI}{l^2} & 4\frac{EI}{l} \end{bmatrix} \quad (7)$$

◦ **Note:** $\mathbf{K}^e$ is a 4×4 matrix, which can be sub-partitioned in four 2×2 nodal matrices

$$\mathbf{K}^e = \left[\begin{array}{cc|cc} \text{x} & \text{x} & \text{x} & \text{x} \\ \text{x} & \text{x} & \text{x} & \text{x} \\ \hline \text{x} & \text{x} & \text{x} & \text{x} \\ \text{x} & \text{x} & \text{x} & \text{x} \end{array}\right] = \begin{bmatrix} \mathbf{K}^e_{ii} & \mathbf{K}^e_{ij} \\ \mathbf{K}^e_{ji} & \mathbf{K}^e_{jj} \end{bmatrix}$$

Beam: direct stiffness formulation I

- **Exercise.** Compute the 6 × 6 stiffness matrix associated to the following set of nodal dofs



- **Exercise.** Investigate on the change of coordinate system for the the 6 × 6 matrix just computed.
Hint. Limiting discussion to a single element (as in figure) assume that we computed the relation:

$$\mathbf{Ku} = \mathbf{F}$$

meant as a local relation, i.e.:

$$\mathbf{K}^l \hat{\mathbf{u}}^l = \mathbf{F}^l \tag{8}$$

where the local vector of dofs $\hat{\mathbf{u}}^l$ is given as:

$$\hat{\mathbf{u}}^l = \left\{ \begin{array}{c} \hat{\mathbf{u}}_1^l \\ \hat{\mathbf{u}}_2^l \end{array} \right\} \quad \text{with} \quad \hat{\mathbf{u}}_1^l = \left\{ \begin{array}{c} \hat{u}_1^l \\ \hat{v}_1^l \\ \hat{\theta}_1^l \end{array} \right\} \text{ and } \hat{\mathbf{u}}_2^l = \left\{ \begin{array}{c} \hat{u}_2^l \\ \hat{v}_2^l \\ \hat{\theta}_2^l \end{array} \right\}$$

Now we can construct the rotation matrix which transform the $\hat{\mathbf{u}}_i$ vectors from a local to a global coordinate system, i.e.

$$\hat{\mathbf{u}}_i^g = \mathbf{R}\hat{\mathbf{u}}_i^l$$

Beam: direct stiffness formulation II

with $\mathbf{R}^{-1} = \mathbf{R}^T$. Accordingly,

$$\hat{\mathbf{u}}^g = \mathbf{T}\hat{\mathbf{u}}^l = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \hat{\mathbf{u}}^l$$

with $\mathbf{T}^{-1} = \mathbf{T}^T$. Now, pre-multiplying relation 8 by $\mathbf{T}$ we obtain

$$\mathbf{T}\mathbf{K}^l\hat{\mathbf{u}}^l = \mathbf{T}\mathbf{F}^l$$

which can be rewritten as:

$$\mathbf{T}\mathbf{K}^l\mathbf{T}^T\mathbf{T}\hat{\mathbf{u}}^l = \mathbf{T}\mathbf{F}^l$$

i.e as:

$$\left(\mathbf{T}\mathbf{K}^l\mathbf{T}^T\right)\left(\mathbf{T}\hat{\mathbf{u}}^l\right) = \mathbf{T}\mathbf{F}^l$$

as well as:

$$\left(\mathbf{T}\mathbf{K}^l\mathbf{T}^T\right)\hat{\mathbf{u}}^g = \mathbf{F}^g$$

where we can identify:

$$\mathbf{K}^g = \mathbf{T}\mathbf{K}^l\mathbf{T}^T$$

- **Exercise.** Solve exercise on page 18 using the 6×6 matrix just computed. [Hint: consider structure without constraint. Construct global stiffness matrix starting from element stiffness matrices. Compute nodal forces. Impose external constraints eliminating specific rows and columns from the system.]

Beam: toward a FEM formulation I

GOAL: compute the same beam stiffness matrix as before, however using a more formal but general line of thinking

1. **Weak formulation of the problem**

   Differential form $\rightarrow$ Integral form

2. **Introduction of approximation fields**

   Integral form $\rightarrow$ Algebraic form

3. **Solution of algebraic problem**

- ⋆ Strong form $\leftrightarrow$ differential form
- ⋆ Weak form $\leftrightarrow$ integral form

- ◦ Differential form is natural format for BVP
- ◦ Weak form often associated to variational principles (necessary to have a functional and to require stationarity)

- ◦ Approximation fields imply problem discretization

- ◦ Algebraic problem (linear or non-linear) easy to solve $\Rightarrow$ matrix inversion

Beam: strong vs weak form I

- **Strong form**

$$EIv^{IV} - q = 0$$

- **Weak form**
  - Multiply differential form by any function w and integrate over domain $[a, b]$

$$\int_a^b \left[w \left(EIv^{IV} - q \right) \right] dx = 0$$

    where w is named weight function
  - Integrate twice by parts

$$\int_a^b \left[w'' EIv'' \right] dx = \int_a^b \left[wq \right] dx - Sw|_a^b + Mw'|_a^b \qquad (9)$$

Beam: strong vs weak form II

- ⋆ Note that
  - equation 9 can be rewritten as

$$a(w, v) = (w, q) - Sw|_a^b + Mw'|_a^b$$

where

$$a(w, v) = \int_a^b \left[w'' EIv''\right] dx \quad , \quad (w, q) = \int_a^b [wq]\, dx$$

  - equation 9 can be rewritten as

$$\boxed{\mathcal{L}_{int} = \mathcal{L}_{ext}} \tag{10}$$

where

$$\begin{cases} \mathcal{L}_{int} = \int_a^b \left[w'' EIv''\right] dx \\ \mathcal{L}_{ext} = \int_a^b [wq]\, dx - Sw|_a^b + Mw'|_a^b \end{cases}$$

- ⋆ Since w is an arbitrary function, Equation 10 is the standard well-know principle of virtual work, i.e. an integral (weak) form expressing beam equilibrium
- In the following neglect boundary terms for notational simplicity

Beam: approximation I

- **Introduce an approximation**

$$\begin{cases} v(x) \approx \sum_{j=1}^{n} N_j(x)\, \hat{v}_j \\ w(x) \approx \sum_{i=1}^{n} N_i(x)\, \hat{w}_i \end{cases}$$

where:

∘ $\hat{v}_j$	**unknown**	parameters	with $j = 1..n$
∘ $\hat{w}_i$	**arbitrary**	parameters	with $i = 1..n$
∘ $N_i(x)$	**known**	functions	with $i = 1..n$

(the so-called shape functions)

- **Note:** $N_i = N_i(x)$ such that

$$\begin{cases} \dfrac{\partial v}{\partial x} \approx \dfrac{\partial}{\partial x} \sum_{j=1}^{n} \hat{v}_j N_j = \sum_{j=1}^{n} \hat{v}_j \dfrac{\partial N_j}{\partial x} \\ \dfrac{\partial w}{\partial x} \approx \dfrac{\partial}{\partial x} \sum_{i=1}^{n} \hat{w}_i N_i = \sum_{i=1}^{n} \hat{w}_i \dfrac{\partial N_i}{\partial x} \end{cases}$$

Beam: approximation I

- In a matrix format

$$\begin{cases} v \approx \sum_{j=1}^{n} N_j \hat{v}_j = \mathbf{N}\hat{\mathbf{v}} \\ w \approx \sum_{i=1}^{n} N_i \hat{w}_i = \mathbf{N}\hat{\mathbf{w}} \end{cases} \qquad \begin{cases} v'' \approx \sum_{j=1}^{n} N_j'' \hat{v}_j = \mathbf{B}\hat{\mathbf{v}} \\ w'' \approx \sum_{i=1}^{n} N_i'' \hat{w}_i = \mathbf{B}\hat{\mathbf{w}} \end{cases}$$

with

$$\begin{cases} \hat{\mathbf{v}} = [\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_n]^T \\ \hat{\mathbf{w}} = [\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_n]^T \\ \mathbf{N} = [N_1, N_2, \ldots, N_n] \\ \mathbf{B} = \left[N_1'', N_2'', \ldots, N_n''\right] \end{cases}$$

Beam: linear system I

- Weak formulation can be rewritten as:

$$\int_a^b [\mathbf{B}\hat{\mathbf{w}} EI \mathbf{B}\hat{\mathbf{v}}]\, dx = \int_a^b [\mathbf{N}\hat{\mathbf{w}} q]\, dx$$

$$\hat{\mathbf{w}}^T \left\{ \int_a^b \left[\mathbf{B}^T EI \mathbf{B}\hat{\mathbf{v}}\right] dx - \int_a^b \left[\mathbf{N}^T q\right] dx \right\} = 0$$

- Arbitrariness of $\hat{\mathbf{w}}$ implies

$$\int_a^b \left[\mathbf{B}^T EI \mathbf{B}\hat{\mathbf{v}}\right] dx = \int_a^b \left[\mathbf{N}^T q\right] dx$$

- Algebraic system

$$\boxed{\mathbf{K}\hat{\mathbf{v}} = \mathbf{f}}$$

with

$$\begin{cases} \mathbf{K} = \int_a^b \left[\mathbf{B}^T EI \mathbf{B}\right] dx \\ \mathbf{f} = \int_a^b \left[\mathbf{N}^T q\right] dx \end{cases}$$

Beam: element point of view I

- Take advantage of additive property of integrals and split integrals over the whole structure as the sum of integrals on the single beams, i.e.

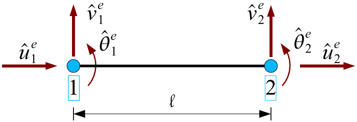
$$\begin{cases} \mathbf{K} = \displaystyle\int_a^b \left[\mathbf{B}^T EI \mathbf{B}\right] dx & = \displaystyle\sum_{e=1}^{nel} \int_{l_e} \left[\mathbf{B}^T EI \mathbf{B}\right] dx & = \displaystyle\sum_{e=1}^{nel} \mathbf{K}^e \\ \mathbf{f} = \displaystyle\int_a^b \left[\mathbf{N}^T q\right] dx & = \displaystyle\sum_{e=1}^{nel} \int_{l_e} \left[\mathbf{N}^T q\right] dx & = \displaystyle\sum_{e=1}^{nel} \mathbf{f}^e \end{cases}$$

where we set:

$$\begin{cases} \mathbf{K}^e = \displaystyle\int_{l_e} \left[\mathbf{B}^T EI \mathbf{B}\right] dx \\ \mathbf{f}^e = \displaystyle\int_{l_e} \left[\mathbf{N}^T q\right] dx \end{cases}$$

Beam: element point of view I

- In general stiffness matrix constructed looking at the single element
- **Global view-point versus Local view-point**
  - Global node numbering vs local node numbering
  - Global dof numbering vs local dof numbering
  - etc. etc.
- **Local view-point** is quite natural for Euler-Bernoulli beam element

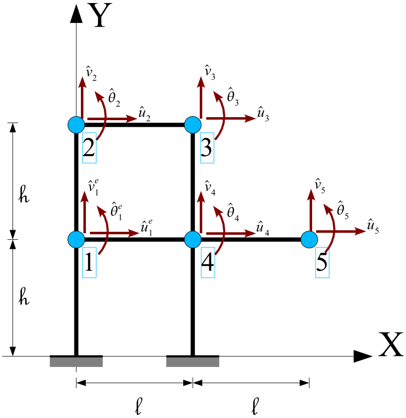


  - 2 nodes
  - 3 dofs per node
  - 6 dofs per element
  - $\mathbf{K}^e[6 \times 6]$
  - $\mathbf{K}^e[6 \times 6]$ can be sub-partitioned in four 3×3 nodal matrices

$$\mathbf{K}^e = \left[\begin{array}{ccc|ccc} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \hline \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \end{array}\right] = \begin{bmatrix} \mathbf{K}_{ii}^e & \mathbf{K}_{ij}^e \\ \mathbf{K}_{ji}^e & \mathbf{K}_{jj}^e \end{bmatrix}$$

Beam: standard element I

- **Global view-point**



- 5 nodes
- 3 dofs per node
- 15 dofs per complete structure
- $\mathbf{K}[15 \times 15]$

Beam: standard element II

- Different dimensions between $\mathbf{K}$ and $\mathbf{K}^e$

$$\mathbf{K}[15 \times 15] \quad , \quad \mathbf{K}^e[6 \times 6]$$

- To manage the different matrix dimensions, convert summation process into the so-called **assembly process**

$$\mathbf{K} = \mathbf{A}_{e=1}^{nel} \mathbf{K}^e$$

◦ **Assembly process**

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11}^{e=1} & \dots & \dots & \mathbf{K}_{14}^{e=1} & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{K}_{41}^{e=1} & \dots & \dots & \mathbf{K}_{44}^{e=1} & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{11}^{e=2} & \mathbf{K}_{12}^{e=2} & \dots & \dots & \dots \\ \mathbf{K}_{21}^{e=2} & \mathbf{K}_{22}^{e=2} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} + \dots$$

$$= \begin{bmatrix} \mathbf{K}_{11}^{e=1} + \mathbf{K}_{11}^{e=2} & \mathbf{K}_{12}^{e=2} & \dots & \mathbf{K}_{14}^{e=1} & \dots \\ \mathbf{K}_{21}^{e=2} & \mathbf{K}_{22}^{e=2} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{K}_{41}^{e=1} & \dots & \dots & \mathbf{K}_{44}^{e=1} & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} + \dots$$

Beam: standard element I

- **Key point:** construction of elementary stiffness and load matrices

$$\begin{cases} \mathbf{K}^e = \int_{l_e} \left[\mathbf{B}^T EI \mathbf{B} \right] dx \\ \mathbf{f}^e = \int_{l_e} \left[\mathbf{N}^T q \right] dx \end{cases}$$

- Start focusing only on the transverse displacement-bending displacement fields
- To do so, we need to define appropriate approximation functions:

$$\begin{cases} v = \mathbf{N}\hat{\mathbf{v}} \\ w = \mathbf{N}\hat{\mathbf{w}} \end{cases} \quad \text{s.t.} \quad \begin{cases} v'' = \mathbf{B}\hat{\mathbf{v}} \\ w'' = \mathbf{B}\hat{\mathbf{w}} \end{cases}$$

- **Question:** how to construct $\mathbf{N}$???

Beam: standard element I

- Standard expression for the deflection:

$$v = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

or in matrix form:

$$v = \mathbf{M}\mathbf{a}$$

with

$$\mathbf{M} = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix}$$

$$\mathbf{a} = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \end{bmatrix}^T$$

- How to compute **a** in terms of "nodal" parameters $\hat{\mathbf{v}}$??

★ Need to riparametrize **a** in terms of the nodal dof for the beam element $\hat{\mathbf{v}}$, i.e.

$$\hat{\mathbf{v}} = \left\{ \hat{v}_1 \quad \hat{\theta}_1 \quad \hat{v}_2 \quad \hat{\theta}_2 \right\}^T$$

★ Impose some specific conditions on the deflection expression v, i.e.

$$\begin{cases} v(0) = \hat{v}_1 \\ v'(0) = \hat{\theta}_1 \end{cases} , \qquad \begin{cases} v(l) = \hat{v}_2 \\ v'(l) = \hat{\theta}_2 \end{cases}$$

Beam: standard element I

- We need to express $\mathbf{a}$ in terms of $\hat{\mathbf{v}}$:

$$
\begin{cases}
v(0) = a_0 & = \hat{v}_1 \\
v'(0) = a_1 & = \hat{\theta}_1 \\
v(l) = a_0 + a_1 l + a_2 l^2 + a_3 l^3 & = \hat{v}_2 \\
v'(l) = a_1 + 2 a_2 l + 3 a_3 l^2 & = \hat{\theta}_2
\end{cases}
$$

- In matrix form:

$$
\mathbf{C}\mathbf{a} = \hat{\mathbf{v}} \quad \text{with} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & l & l^2 & l^3 \\ 0 & 1 & 2l & 3l^2 \end{bmatrix}
$$

- Inverting the above relation :

$$
\mathbf{C}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\dfrac{3}{l^2} & -\dfrac{2}{l} & \dfrac{3}{l^2} & -\dfrac{1}{l} \\ \dfrac{2}{l^3} & \dfrac{1}{l^2} & -\dfrac{2}{l^3} & \dfrac{1}{l^2} \end{bmatrix}
$$

Beam: standard element I

- Concluding:

$$v = \mathbf{Ma} = \mathbf{MC}^{-1}\hat{\mathbf{v}}$$

i.e.

$$v = \mathbf{N}\hat{\mathbf{v}}$$

with

$$\mathbf{N} = \mathbf{MC}^{-1} = [N_1 \quad N_2 \quad N_3 \quad N_4]$$

or in a more explicit format:

$$\begin{cases} N_1 = 1 - 3\dfrac{x^2}{l^2} + 2\dfrac{x^3}{l^3} \\ N_2 = x - 2\dfrac{x^2}{l} + \dfrac{x^3}{l^2} \end{cases} , \quad \begin{cases} N_3 = 3\dfrac{x^2}{l^2} - 2\dfrac{x^3}{l^3} \\ N_4 = \dfrac{x^3}{l^2} - \dfrac{x^2}{l} \end{cases} \tag{11}$$

- ☐ **Exercise.** Verify correctness of matrices **C** and $\mathbf{C}^{-1}$.
- ☐ **Exercise.** Verify correctness of shape functions N_i using expression 11.
- ☐ **Exercise.** Verify correctness of shape functions N_i solving the corresponding beam boundary value problems.
- ☐ **Exercise.** Given shape functions N_i, compute element stiffness matrix and compare it with matrix 7.

Beam: a matlab code to compute standard element stiffness I

```
% Compute stiffness matrix associated to Euler-Bernoulli beam theory

clear all

syms x
syms a0 a1 a2 a3           % coefficients of polynomial cubic
syms v1 theta1 v2 theta2   % nodal dofs
syms l                     % beam length
syms KK                    % stiffness matrix

% Elastica expression in terms of cubic constants

v = a0 + a1*x + a2*x^2 + a3*x^3     % elastica transv.displ.
theta = diff(v,x)                   % elastica rotation

% Convert elastica cubic constants in terms of nodal dofs

eq1=subs(v,x,0)-v1
eq2=subs(theta,x,0)-theta1
eq3=subs(v,x,l)-v2
eq4=subs(theta,x,l)-theta2

[a0,a1,a2,a3] = solve(eq1,eq2,eq3,eq4,a0,a1,a2,a3)

% Elastica expression in terms of nodal dofs

v = a0 + a1*x + a2*x^2 + a3*x^3     % elastica expression
```

Beam: a matlab code to compute standard element stiffness II

```
% Compute shape functions

N1=subs(v,{v1,theta1,v2,theta2},[1,0,0,0]) % shape function for v1-dof
N2=subs(v,{v1,theta1,v2,theta2},[0,1,0,0]) % shape function for theta1-dof
N3=subs(v,{v1,theta1,v2,theta2},[0,0,1,0]) % shape function for theta1-dof
N4=subs(v,{v1,theta1,v2,theta2},[0,0,0,1]) % shape function for theta1-dof

% Compute stiffness matrix terms

N1_d2 = simplify(diff(N1,x,2))      % take 2nd derivative of N1
N2_d2 = simplify(diff(N2,x,2))      % take 2nd derivative of N2
N3_d2 = simplify(diff(N3,x,2))      % take 2nd derivative of N3
N4_d2 = simplify(diff(N4,x,2))      % take 2nd derivative of N4


KK(1,1) = int(N1_d2*N1_d2,x,0,l);   % compute v1-v1     stiffness
KK(1,2) = int(N1_d2*N2_d2,x,0,l);   % compute v1-theta1 stiffness
KK(1,3) = int(N1_d2*N3_d2,x,0,l);   % compute v1-v2     stiffness
KK(1,4) = int(N1_d2*N4_d2,x,0,l);   % compute v1-theta2 stiffness

KK
```

Beam: a matlab code implementing EB beam FE I

```
function displ = frame2d_lin(flag)

%==================================================================
% 2D FRAME FINITE ELEMENT PROGRAM
% (Euler-Bernoulli and Timoshenko linear beam theories)
% (by A. Reali and F. Auricchio)
%------------------------------------------------------------------
% SYNTAX:        displ = frame2d_lin(flag)
%------------------------------------------------------------------
% INPUT
% flag=0 => Euler-Bernoulli beam
% flag=1 => Timoshenko beam (linear interpolation)
% flag=2 => Timoshenko beam (linear+linked interpolation)
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
% nodes.dat:
%                ...
%                x y bc
%                ...
%
% (bc=0 => free, bc=1 => clamped)
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
% elements.dat:
%               Euler-Bernoulli beam:
%               ...
%               node1 node2 E A I
%               ...
%
```

Beam: a matlab code implementing EB beam FE II

```
%                Timoshenko beam:
%                ...
%                node1 node2 E A I G k
%                ...
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
% loads.dat:
%                ...
%                loadednode Fx Fy Mz
%                ...
%----------------------------------------------------------------
% OUTPUT
% displ:
%                ...
%                ux uy thetaz
%                ...
%================================================================

% load mesh data and material properties
load nodes.dat
load elements.dat
load loads.dat

echo = 0;     % set input echo: 0=>off 1=>on

% print mesh data and material parameters
if (echo == 1)
    nodes
    elements
```

Beam: a matlab code implementing EB beam FE III

```
    loads
end

nnod  = size(nodes,1);
nel   = size(elements,1);
nload = size(loads,1);

% find active nodes
active = find(nodes(:,3) == 0);
nact = length(active);
ndof = 3*nact;

if (echo == 1)
    active     % print active nodes
end

% inizializations
k_gl  = zeros(ndof);
f_gl  = zeros(ndof,1);
displ = zeros(nnod,3);

for i = 1:nload     % loop over loaded nodes to form r-h-s vector
    ii = find(active == loads(i,1));
    if isempty(ii)== 0
        f_gl((3*ii-2):(3*ii)) = loads(i,2:4);
    end
end
```

Beam: a matlab code implementing EB beam FE IV

```
for i = 1:nel     % loop over elements
    nodes_el = nodes(elements(i,1:2),1:2) ;
    % compute element stiffness
    if flag == 0
        k_el = EB_k_el(elements(i,3:5),nodes_el);
    else
        k_el = T_k_el(elements(i,3:7),nodes_el,flag);
    end
    % assemble into global stiffness
    k_gl = k_assembl(k_el,k_gl,elements(i,1:2),active);
end

u_gl = k_gl\f_gl;     % solve the linear system by gaussian elimination
displ(active,:) = reshape(u_gl,3,nact)';
```

Beam: a matlab code implementing EB beam FE I

```
function k_el = EB_k_el(element,nodes)
% compute the Euler-Bernoulli element stiffness matrix

% set element nodes and material properties
P1 = nodes(1,:);
P2 = nodes(2,:);
E  = element(1);
A  = element(2);
I  = element(3);
%G = element(4);
%k = element(5);

L     = norm(P2-P1);    % compute beam length
e1    = (P2 - P1)'/L;   % compute 1st local axis coincident with beam axis
e1(3) = 0;
e3    = [0;0;1];        % set 3rd local axis coincident with global z axis
e2    = cross(e3,e1);   % compute second local axis as e3xe1

% construct element submatrices
k_axial = E*A/L*[1 -1; -1 1];
a1 = 12/L^3;
a2 = 6/L^2;
a3 = 4/L;
a4 = a3/2;
k_bend = E*I*[a1  a2 -a1  a2;
              a2  a3 -a2  a4;
             -a1 -a2  a1 -a2;
```

Beam: a matlab code implementing EB beam FE II

```
             a2  a4 -a2  a3];

% construct the element stiffness
k_el = zeros(6);
k_el([1 4],[1 4]) = k_axial;
k_el([2 3 5 6],[2 3 5 6]) = k_bend;

% rotate from local to global axes
e1 = e1(1:2);
e2 = e2(1:2);
E1 = [1 0];
E2 = [0 1];
R  = [E1*e1 E2*e1 0; E1*e2 E2*e2 0; 0 0 1];
k_el = [R zeros(3); zeros(3) R]'*k_el*[R zeros(3); zeros(3) R];
```

Beam: a matlab code implementing EB beam FE I

```
function k_gl = k_assembl(k_el,k_gl,element,active)
% assemble the local matrix k_el into the global one k_gl

% check if the element nodes are active or constrained
i1 = find(active == element(1));
i2 = find(active == element(2));

if isempty(i1)
   index1_gl  = [];
   index1_loc = [];
else
   index1_gl  = (3*i1-2):(3*i1);
   index1_loc = 1:3;
end

if isempty(i2)
   index2_gl  = [];
   index2_loc = [];
else
   index2_gl  = (3*i2-2):(3*i2);
   index2_loc = 4:6;
end

% set global and local indices
index_gl  = [index1_gl,index2_gl];
index_loc = [index1_loc,index2_loc];
```

Beam: a matlab code implementing EB beam FE II

```
% assemble k_el into k_gl
k_gl(index_gl,index_gl) = k_gl(index_gl,index_gl) +...
                          k_el(index_loc,index_loc);
```

Beam: standard element I

- **Advantages:**
  - For the case of concentrated loads (forces and moments) the presented element gives exact solutions
- **Disadvantages:**
  - Require the use of high-order shape functions (continuous with the first derivative) $\Rightarrow$ difficult to generalize to plate / shell problems
  - Impossible to extend to the case of non-linear problems

☐ **Exercise.** Consider the case of a cantilever beam loaded with a end moment. Compute the Euler-Bernoulli analytical solution in terms of displacement and rotation and compare it with the finite-element solution for the case of 1 element and for the case of 10 elements.

☐ **Exercise.** Consider the case of a cantilever beam loaded with a end force. Compute the Euler-Bernoulli analytical solution in terms of displacement and rotation and compare it with the finite-element solution for the case of 1 element and for the case of 10 elements.

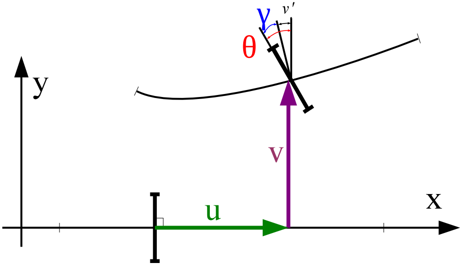
☐ **Exercise.** Consider the case of a cantilever beam loaded with a distributed transverse force. Compute the Euler-Bernoulli analytical solution in terms of displacement and rotation and compare it with the finite-element solution for the case of 1 element and for the case of 10 elements.

Timoshenko beam: differential eqns I

- **Timoshenko beam:** beam with shear deformation

Plane sections normal to beam axis remain plane but not necesseraly normal to axis during the deformation

$$\gamma \neq 0 \qquad \Rightarrow \qquad \theta \neq v'$$



Timoshenko beam: weak form I

- To obtain an integral (weak) form for Timoshenko beam use a potential energy approach

- **Total Potential Energy** obtained summing energy associated to bending, energy associated to shear and potential Π^{ext} associated to external forces

$$\Pi(v,\theta) = \frac{1}{2}\int_l \left[EI\chi^2\right] dx + \frac{1}{2}\int_l \left[\kappa GA\gamma^2\right] dx - \Pi^{ext}$$

  - We recall that

$$\begin{cases} \chi(\theta) = \theta' \\ \gamma(v,\theta) = v' - \theta \end{cases}$$

  hence

$$\Pi = \Pi(\chi(\theta), \gamma(v,\theta)) = \Pi(v,\theta)$$

- Equilibrium equations obtained requiring stationarity of total potential energy
  - Total potential energy is a functional, i.e.

$$\Pi : \{v(x), \theta(x)\} \rightarrow \Pi \in \mathcal{R} \quad \text{with} \quad \begin{cases} v(x) : x \in \mathcal{R} \rightarrow v(x) \in \mathcal{R} \\ \theta(x) : x \in \mathcal{R} \rightarrow \theta(x) \in \mathcal{R} \end{cases}$$

Stationarity conditions I

- **Real function** $f : x \in \mathcal{R} \rightarrow \mathcal{R}$
- Stationarity of f obtained requiring that the function derivative is null, i.e.

$$df = 0 \qquad \Leftrightarrow \qquad \frac{df}{dx} = 0$$

- *Derivative definition*. The derivative f' of a function f in a point x_0 is defined as:

$$f'(x_0) = \left.\frac{df}{dx}\right|_{x_0} = \lim_{\epsilon \rightarrow 0} \frac{f(x_0 + \epsilon) - f(x_0)}{\epsilon}$$

- *Fundamental lemma of differentiation*. Suppose that f has a derivative at x_0. Then, there exists a function η (defined around 0) such that

$$f(x_0 + h) - f(x_0) = \left[f'(x_0) + \eta(\epsilon)\right] \epsilon$$

  Function η is continuous with $\eta(0) = 0$.

⋆ The previous lemma states that a differentiable function can be approximated by a linear function whose slope is the derivative

Stationarity conditions I

- **Real function** $f : \mathbf{x} = \{x_1, x_2\} \in \mathcal{R}^2 \to \mathcal{R}$
- Stationarity of f obtained requiring that the function differential is null, i.e.

$$df = 0 \qquad \Leftrightarrow \qquad \frac{\partial f}{\partial x_1} = \frac{\partial f}{\partial x_2} = 0$$

- *Directional derivative*. Directional derivatives $f_{,1}$ and $f_{,2}$ are defined as:

$$\begin{cases} f_{,1} = \dfrac{\partial f}{\partial x_1} = \lim\limits_{\epsilon \to 0} \dfrac{f(x_1 + \epsilon, x_2) - f(x_1, x_2)}{\epsilon} \\ f_{,2} = \dfrac{\partial f}{\partial x_2} = \lim\limits_{\epsilon \to 0} \dfrac{f(x_1, x_2 + \epsilon) - f(x_1, x_2)}{\epsilon} \end{cases}$$

- *Fundamental lemma of differentiation*. Suppose that f has continuous directional derivatives in $\mathbf{x}$. Then, f is continuous; moreover, $\exists$ functions η_1, η_2 continuous at 0 with $\eta_1(0) = \eta_2(0) = 0$ s.t.

$$f(\mathbf{x} + \mathbf{h}) - f(\mathbf{x}) = \sum_{i=1}^{2} \left[f_{,i}(\mathbf{x}) + \eta_i(\mathbf{h}) \right] h_i$$

with $\mathbf{h} = \{h_1, h_2\}$.

Stationarity conditions I

⋆ The above lemma states that every function with continuous directional derivatives can be approximated by a linear function

$$f(\mathbf{x}+\mathbf{h}) - f(\mathbf{x}) = \sum_{i=1}^{2} \left[f_{,i}(\mathbf{x}) + \eta_i(\mathbf{h}) \right] h_i$$

⋆ Moreover:

$$\begin{aligned} f(\mathbf{x}+\mathbf{h}) &= f(\mathbf{x}) + \sum_{i=1}^{2} \left[f_{,i}(\mathbf{x}) h_i \right] + \sum_{i=1}^{2} \left[\eta_i(\mathbf{h}) h_i \right] \\ &= f(\mathbf{x}) + Df(\mathbf{x})\,[\mathbf{h}] + o(\mathbf{h}) \end{aligned}$$

- Differential. A function f is differentiable if $\exists$ a linear transformation $Df : \mathcal{R}^2 \to \mathcal{R}^2$ s.t.

$$f(\mathbf{x}+\mathbf{h}) = f(\mathbf{x}) + Df(\mathbf{x})\,[\mathbf{h}] + o(\mathbf{h})$$

as $\mathbf{h} \to 0$

- If the differential $Df(\mathbf{x})$ exists, it is unique and defined as:

$$Df(\mathbf{x})[\mathbf{h}] = \lim_{\alpha \to 0} \frac{f(\mathbf{x}+\alpha\mathbf{h}) - f(\mathbf{x})}{\alpha} = \left[\frac{d}{d\alpha} f(\mathbf{x}+\alpha\mathbf{h}) \right]_{\alpha=0}$$

Stationarity conditions I

- **Functional** F is a real valued function whose domain is a set of real functions

$$F : \mathbf{v}(x) \rightarrow \mathcal{R}$$

In general

$$F : \mathbf{v} \in \mathbf{V} \rightarrow \mathcal{R}$$

with $\mathbf{V}$ vector space

- Stationarity of functional F obtained requiring variations of F (or the differential) to be identically null, i.e.

$$DF(\mathbf{v})[\mathbf{u}] = 0 \tag{12}$$

for all possible variations $\mathbf{u}$!!!

  - Recall that

$$DF(\mathbf{v})[\mathbf{u}] = \lim_{\alpha \rightarrow 0} \frac{F(\mathbf{v} + \alpha \mathbf{u}) - F(\mathbf{v})}{\alpha} = \left[\frac{d}{d\alpha} F(\mathbf{v} + \alpha \mathbf{u}) \right]_{\alpha = 0}$$

  - Sometimes it is common the following notation

$$DF(\mathbf{v})[\delta \mathbf{v}]$$

with $\delta \mathbf{v}$ indicated as variations

Timoshenko beam: weak form I

- Taking the variations we get the weak form associated to the Timoshenko beam

$$d\Pi(v,\theta)[\delta v,\delta\theta] = \int_l [\delta\chi EI\chi]\, dx + \int_l [\delta\gamma\kappa GA\gamma]\, dx - \delta\Pi^{ext} = 0 \tag{13}$$

where

$$\begin{cases} \chi = \theta' \\ \delta\chi = [\delta\theta]' \\ \gamma = v' - \theta \\ \delta\gamma = [\delta v]' - \delta\theta \end{cases}$$

- Functions $\delta\theta$ and δv play the role of weight functions
- Since Equation 13 has to be satisfied for all possible variations, we can write

$$\begin{cases} d\Pi(v,\theta)[0,\delta\theta] = 0 \\ d\Pi(v,\theta)[\delta v, 0] = 0 \end{cases}$$

Timoshenko beam: weak form I

$$\begin{cases} \int_l \left[\delta\theta' EI \theta' \right] dx - \int_l \left[\delta\theta \kappa GA \left(v' - \theta \right) \right] dx = 0 \\ \int_l \left[\delta v' \kappa GA \left(v' - \theta \right) \right] dx - \delta\Pi^{ext} = 0 \end{cases} \qquad (14)$$

- We can start from Equation 14 to develop a Timoshenko FE beam model

☐ **Exercise.** Discuss that Equation 13_1 can be interpreted as a rotational equilibrium equation and Equation 13_2 can be interpreted as a translational equilibrium equation. In particular, you can rewrite them as:

$$\begin{cases} \int_l \left[\delta\theta' M \right] dx - \int_l \left[\delta\theta S \right] dx = 0 \\ \int_l \left[\delta v' S \right] dx - \delta\Pi^{ext} = 0 \end{cases}$$

recalling that:

$$M = EI\theta' \qquad , \qquad S = \kappa GA \left(v' - \theta \right)$$

☐ **Exercise.** Derive the strong form of the equations associated to the functional introduced above. Do they correspond to a Timoshenko beam problem?

Timoshenko beam: approximation I

- Introduce an approximation

$$\begin{cases} v \approx \sum_{a=1}^{nv} N_a^v \hat{v}_a \\ \theta \approx \sum_{a=1}^{n\theta} N_a^\theta \hat{\theta}_a \end{cases} \qquad \begin{array}{l} \delta v \approx \sum_{a=1}^{nv} N_a^v \delta\hat{v}_a \\ \delta\theta \approx \sum_{a=1}^{n\theta} N_a^\theta \delta\hat{\theta}_a \end{array}$$

  where nv and $n\theta$ can differ

- Accordingly:

$$\begin{cases} \chi = \theta' \approx \left[\sum_{a=1}^{n\theta} N_a^\theta \hat{\theta}_a \right]' = \sum_{a=1}^{n\theta} \left(N_a^\theta \right)' \hat{\theta}_a \\ \gamma = v' - \theta \approx \left[\sum_{a=1}^{nv} N_a^v \hat{v}_a \right]' - \left[\sum_{a=1}^{n\theta} N_a^\theta \hat{\theta}_a \right] \\ \quad = \left[\sum_{a=1}^{nv} (N_a^v)' \hat{v}_a \right] - \left[\sum_{a=1}^{n\theta} N_a^\theta \hat{\theta}_a \right] \end{cases}$$

- Similar developments for weight functions $\delta\chi$ and $\delta\gamma$

Timoshenko beam: approximation I

- In a matrix format

$$\begin{cases} v \approx \sum_{a=1}^{nv} N_a^v \hat{v}_a = \mathbf{N}^v \hat{\mathbf{v}} \\ \theta \approx \sum_{a=1}^{n\theta} N_a^\theta \hat{\theta}_a = \mathbf{N}^\theta \hat{\boldsymbol{\theta}} \end{cases}$$

with

$$\begin{cases} \hat{\mathbf{v}} = [\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_{nv}]^T \\ \hat{\boldsymbol{\theta}} = \left[\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_{n\theta}\right]^T \\ \mathbf{N}^v = [N_1^v, N_2^v, \ldots, N_{nv}^v] \\ \mathbf{N}^\theta = \left[N_1^\theta, N_2^\theta, \ldots, N_{n\theta}^\theta\right] \end{cases}$$

- Note that
  - for notational simplicity we indicate with $\hat{\mathbf{v}}$ and $\hat{\boldsymbol{\theta}}$ the nodal dofs corresponding respectively to the transverse displacement and to the rotation
  - similar considerations apply to $\delta\hat{\mathbf{v}}$ and $\delta\hat{\boldsymbol{\theta}}$

Timoshenko beam: approximation I

- Accordingly:

$$\begin{cases} \chi = \theta' \approx \displaystyle\sum_{a=1}^{n\theta} \left(N_a^\theta\right)' \hat{\theta}_a = \mathbf{B}^\theta \hat{\boldsymbol{\theta}} \\ \gamma = v' - \theta \approx \left[\displaystyle\sum_{a=1}^{nv} (N_a^v)' \hat{v}_a\right] - \left[\displaystyle\sum_{a=1}^{n\theta} N_a^\theta \hat{\theta}_a\right] = \mathbf{B}^v \hat{\mathbf{v}} - \mathbf{N}^\theta \hat{\boldsymbol{\theta}} \end{cases}$$

with

$$\begin{cases} \hat{\mathbf{v}} = [\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_{nv}]^T \\ \hat{\boldsymbol{\theta}} = \left[\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_{n\theta}\right]^T \\ \mathbf{N}^\theta = \left[N_1^\theta, N_2^\theta, \ldots, N_{n\theta}^\theta\right] \\ \mathbf{B}^\theta = \left[\left(N_1^\theta\right)', \left(N_2^\theta\right)', \ldots, \left(N_{n\theta}^\theta\right)'\right] \\ \mathbf{B}^v = \left[(N_1^v)', (N_2^v)', \ldots, (N_{nv}^v)'\right] \end{cases}$$

- Similar developments for the weight functions:

$$\begin{cases} \delta\chi = \mathbf{B}^\theta \delta\hat{\boldsymbol{\theta}} \\ \delta\gamma = \mathbf{B}^v \delta\hat{\mathbf{v}} - \mathbf{N}^\theta \delta\hat{\boldsymbol{\theta}} \end{cases}$$

Timoshenko beam: approximation I

- Introduce approximations in the weak form 13 (and not – also if equivalent – in weak form variation 14) :

$$\int_l \left[\left(\mathbf{B}^\theta \delta \hat{\boldsymbol{\theta}} \right) EI \mathbf{B}^\theta \hat{\boldsymbol{\theta}} \right] dx + \int_l \left[\left(\mathbf{B}^v \delta \hat{\mathbf{v}} - \mathbf{N}^\theta \delta \hat{\boldsymbol{\theta}} \right) \kappa GA \left(\mathbf{B}^v \hat{\mathbf{v}} - \mathbf{N}^\theta \hat{\boldsymbol{\theta}} \right) \right] dx - \delta \Pi^{ext} = 0$$

- Rearranging terms, we get:

$$\begin{aligned} &\left(\delta \hat{\boldsymbol{\theta}} \right)^T \int_l \left[\left(\mathbf{B}^\theta \right)^T EI \mathbf{B}^\theta \hat{\boldsymbol{\theta}} \right] dx - \\ &\left(\delta \hat{\boldsymbol{\theta}} \right)^T \int_l \left[\left(\mathbf{N}^\theta \right)^T \kappa GA \left(\mathbf{B}^v \hat{\mathbf{v}} - \mathbf{N}^\theta \hat{\boldsymbol{\theta}} \right) \right] dx + \\ &\left(\delta \hat{\mathbf{v}} \right)^T \int_l \left[\left(\mathbf{B}^v \right)^T \kappa GA \left(\mathbf{B}^v \hat{\mathbf{v}} - \mathbf{N}^\theta \hat{\boldsymbol{\theta}} \right) \right] dx - \delta \Pi^{ext} = 0 \end{aligned}$$

- Arbitrariness of $\delta \hat{\boldsymbol{\theta}}$ and of $\delta \hat{\mathbf{v}}$ imply:

$$\begin{cases} \int_l \left[\left(\mathbf{B}^\theta \right)^T EI \mathbf{B}^\theta \hat{\boldsymbol{\theta}} \right] dx - \int_l \left[\left(\mathbf{N}^\theta \right)^T \kappa GA \left(\mathbf{B}^v \hat{\mathbf{v}} - \mathbf{N}^\theta \hat{\boldsymbol{\theta}} \right) \right] dx - .... = \mathbf{0} \\ \int_l \left[\left(\mathbf{B}^v \right)^T \kappa GA \left(\mathbf{B}^v \hat{\mathbf{v}} - \mathbf{N}^\theta \hat{\boldsymbol{\theta}} \right) \right] dx - ..... = \mathbf{0} \end{cases}$$

Timoshenko beam: approximation I

- This is a set of two matrix equations

$$\mathbf{K}\hat{\mathbf{u}} = \mathbf{f}$$

more explicitly

$$\begin{bmatrix} \mathbf{K}^{\theta\theta} & \mathbf{K}^{\theta v} \\ \left(\mathbf{K}^{\theta v}\right)^T & \mathbf{K}^{vv} \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{\theta}} \\ \hat{\mathbf{v}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ .... \end{Bmatrix}$$

where

$$\begin{cases} \mathbf{K}^{\theta\theta} = & \int_l \left[\left(\mathbf{B}^\theta\right)^T EI \mathbf{B}^\theta \right] dx + \int_l \left[\left(\mathbf{N}^\theta\right)^T \kappa GA \mathbf{N}^\theta \right] dx \\ \mathbf{K}^{\theta v} = & - \int_l \left[\left(\mathbf{N}^\theta\right)^T \kappa GA \mathbf{B}^v \right] dx \\ \mathbf{K}^{vv} = & \int_l \left[\left(\mathbf{B}^v\right)^T \kappa GA \mathbf{B}^v \right] dx \end{cases}$$

- As usual we can introduce element point of view $\rightarrow$ element stiffness matrices

Timoshenko beam I

- **Advantages:**
  - The element takes into account shear deformation $\Rightarrow$ possible to study a wider range of problems
  - Require the use of low-order shape functions (discontinuous first derivatives) $\Rightarrow$ easy to generalize to plate / shell problems
- **Disadvantages:**
  - For the case of concentrated loads (forces and moments) in general Timoshenko elements do not give exact solutions
  - Impossible to extend to the case of non-linear problems

Timoshenko beam: linear approx I

- **Simplest choice**
  - two node element
  - two dofs per node: 1 transv. displ. + 1 rotation
  - linear approximations for all the fields, i.e.

$$\begin{cases} v = \mathbf{N}\hat{\mathbf{v}} \\ \theta = \mathbf{N}\hat{\boldsymbol{\theta}} \end{cases}$$

  with

$$\mathbf{N} = \left[1 - \frac{x}{l} \quad , \quad \frac{x}{l}\right]$$

- Accordingly:

$$\mathbf{B} = \left[-\frac{1}{l} \quad , \quad \frac{1}{l}\right]$$

☐ **Exercise.** Compute in close form the element stiffness matrix for the Timoshenko beam considered above.

☐ **Exercise.** Code the element and test it solving cantilever beam problems either with an end moment or with an end load.

☐ **Exercise.** Consider the case of a cantilever beam loaded with a end moment and the case of a cantilever beam loaded with a end force. Compute the Euler-Bernoulli analytical solution in terms of displacement and rotation and compare it with the finite-element solution for the case of 1 element, 10 elements, 100 elements. Discuss the results. Diagram all the quantities of interest and discuss possible requirements on the finite element scheme.

☐ **Exercise.** Consider a series of cantilever beam problems in which the beam is progressively more and more slender. Load the cantilevers first with an end moment and then with an end load.

Timoshenko beam: a matlab code to compute standard element stiffness I

```
function k_el = T_k_el(element,nodes,flag)
% compute the Timoshenko element stiffness matrix
% flag=1 => linear interpolations
% flag=2 => linear+linked interpolations (not active, so far!)

% set element nodes and material properties
P1 = nodes(1,:);
P2 = nodes(2,:);
E  = element(1);
A  = element(2);
I  = element(3);
G  = element(4);
k  = element(5);

L     = norm(P2-P1);    % compute beam length
e1    = (P2 - P1)'/L;   % compute 1st local axis coincident with beam axis
e1(3) = 0;
e3    = [0;0;1];        % set 3rd local axis coincident with global z axis
e2    = cross(e3,e1);   % compute second local axis as e3xe1

% construct element submatrices
k_uu = E*A/L*[1 -1; -1 1];
k_vv = k*G*A/L*[1 -1; -1 1];
k_tt = E*I/L*[1 -1; -1 1] + k*G*A*L/3*[1 .5; .5 1];
k_tv = k*G*A*[.5 -.5; .5 -.5];

% construct the element stiffness
```

Timoshenko beam: a matlab code to compute standard element stiffness II

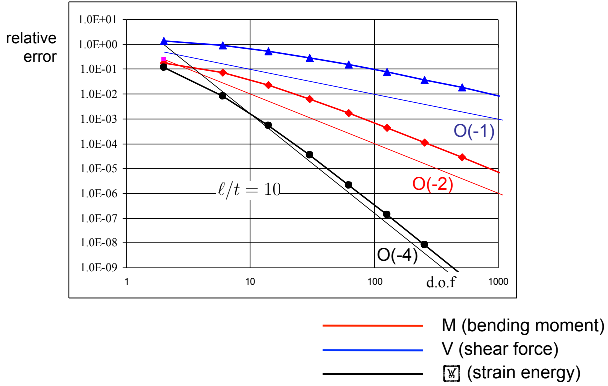
```
k_el = zeros(6);
k_el([1 4],[1 4]) = k_uu;
k_el([2 5],[2 5]) = k_vv;
k_el([3 6],[3 6]) = k_tt;
k_el([3 6],[2 5]) = k_tv;
k_el([2 5],[3 6]) = k_tv';

% rotate from local to global axes
e1 = e1(1:2);
e2 = e2(1:2);
E1 = [1 0];
E2 = [0 1];
R  = [E1*e1 E2*e1 0; E1*e2 E2*e2 0; 0 0 1];
k_el = [R zeros(3); zeros(3) R]'*k_el*[R zeros(3); zeros(3) R];
```

Timoshenko beam: locking problems I

Numerical Experiments

thick beam, Bernoulli beam elements

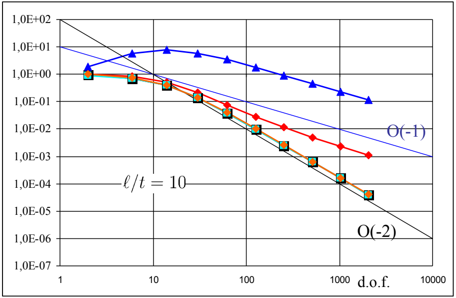


Timoshenko beam: locking problems II

Numerical Experiments

thick beam, Timoshenko beam elements

relative error

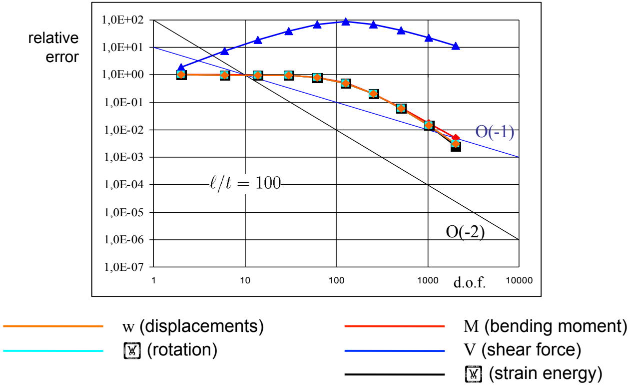


w (displacement)
(rotation)
M (bending moment)
V (shear force)
(strain energy)

Timoshenko beam: locking problems III

Numerical Experiments

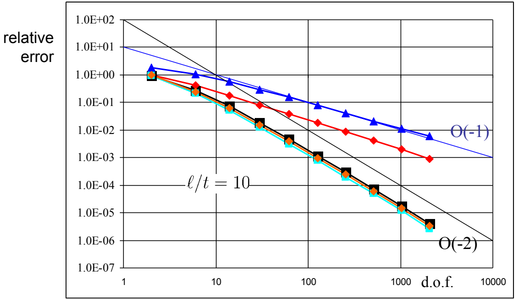
thin beam, Timoshenko beam elements



Manfred Bischoff, IBB, Uni Stuttgart | Structural Finite Elements | April 16, 2012 | 21/40

Numerical Experiments

thin beam, reduced integration (Timoshenko)



w (displacement)

(rotation)

M (bending moment)

V (shear force)

(strain energy)

Manfred Bischoff, IBB, Uni Stuttgart | Structural Finite Elements | April 16, 2012 | 22/40

Timoshenko beam: locking problems V

- Compute the shear deformation for the beam under investigation

$$\begin{aligned}\gamma =& \mathbf{B}\hat{\mathbf{v}} - \mathbf{N}\hat{\boldsymbol{\theta}} \\ =& \left(-\frac{1}{l}\hat{v}_1 + \frac{1}{l}\hat{v}_2\right) - \left[\left(1 - \frac{x}{l}\right)\hat{\theta}_1 + \frac{x}{l}\hat{\theta}_2\right] \\ =& -\frac{x}{l}\left(\hat{\theta}_2 - \hat{\theta}_1\right) - \left[\frac{\hat{v}_1 - \hat{v}_2}{l} + \hat{\theta}_1\right]\end{aligned}$$

- In a more simple form

$$\gamma = c_1 x + c_0$$

  with c_1 and c_0 constants

- In the limiting case of thin beam, Timoshenko model should recover Euler Bernoulli model

$$\gamma \to 0 \qquad \Rightarrow \qquad \begin{cases} c_1 = 0 \\ c_0 = 0 \end{cases}$$

- These conditions result in a *over-stiffening* of the element, i.e. **LOCKING**
- The locking is due to the presence of constraint which we are not properly satisfied using the chosen interpolations

Timoshenko beam: linear + linked I

- **Second possible choice**
  - two node element
  - two dofs per node: 1 transv. displ. + 1 rotation
  - linear approximation for θ-field
  - linear approximation + **linked term** for v-field

$$\begin{cases} v = \mathbf{N}\hat{\mathbf{v}} + N_b \left(\hat{\theta}_1 - \hat{\theta}_2\right) l \\ \theta = \mathbf{N}\hat{\boldsymbol{\theta}} \end{cases}$$

with

$$\mathbf{N} = \left[\left(1 - \frac{x}{l}\right) \quad , \quad \frac{x}{l}\right]$$

$$N_b = \frac{1}{2}\left(1 - \frac{x}{l}\right)\frac{x}{l}$$

⋆ **Linked** because it links v-field to θ-field

⋆ N_b **bubble function**

Timoshenko beam: linear + linked I

- **Exercise.** Compute γ for the Timoshenko beam considered above and compare it with the case of linear element.
- **Exercise.** Compute in close form the element stiffness matrix for the Timoshenko beam considered above.
- **Exercise.** Code the element and test it solving cantilever beam problems either with an end moment or with an end load.
- **Exercise.** Consider a series of cantilever beam problems in which the beam is progressively slender and slender. Load the cantilevers either with an end moment or with an end load.

Timoshenko beam: mixed approach I

- As integral form we can use a **mixed** functional [Hellinger-Reisser functional]

$$\Pi(v,\theta,S) = \frac{1}{2}\int_l \left[EI\chi^2\right] dx - \frac{1}{2}\int_l \left[(\kappa GA)^{-1} S^2\right] dx + \int_l \left[S\left(v' - \theta\right)\right] dx - \Pi^{ext}$$

where

  - Π^{ext} is the potential associated to the external forces
  - the curvature is always function of the rotation

$$\chi = \theta'$$

☐ **Exercise.** Compute the stationarity conditions for Hellinger-Reissner principle and the corresponding strong form equations. Comment on the obtained strong form conditions.

Timoshenko beam: mixed approach II

- **Simplest choice for mixed functional**

★ piecewise linear (continuous) displacement/rotation fields:

- two node element
- two dofs per node: 1 transv. displ. + 1 rotation

$$\begin{cases} v = \mathbf{N}\hat{\mathbf{v}} \\ \theta = \mathbf{N}\hat{\boldsymbol{\theta}} \end{cases}$$

with

$$\mathbf{N} = \left[1 - \frac{x}{l} \quad , \quad \frac{x}{l} \right]$$

★ constant approximation for shear force:

- one internal node
- one dof per node

$$S = N_S \hat{S} = \hat{S}$$

Timoshenko beam: mixed approach III

- **Exercise.** Compute in close form the element stiffness matrix for the Timoshenko beam considered above. Compare this stiffness matrix with the one obtained from the linked approach.
- **Exercise.** Compute in close form the element stiffness matrix for the Timoshenko beam considered above and perform a static condensation of the local shear dof. Compare this stiffness matrix with the one obtained from the linked approach.
- **Exercise.** Code the element and test it solving cantilever beam problems either with an end moment or with an end load.
- **Exercise.** Consider a series of cantilever beam problems in which the beam is progressively slender and slender. Load the cantilevers either with an end moment or with an end load.

Timoshenko beam: mixed approach & static condensation I

- Write explicitly equation system for mixed formulation:

$$\begin{bmatrix} \mathbf{K}^{\theta\theta} & \mathbf{0} & \mathbf{K}^{\theta S} \\ \mathbf{0} & \mathbf{0} & \mathbf{K}^{vS} \\ \left(\mathbf{K}^{\theta S}\right)^T & \left(\mathbf{K}^{vS}\right)^T & -\mathbf{K}^{SS} \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{\theta}} \\ \hat{\mathbf{v}} \\ \hat{\mathbf{S}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \cdots \\ \mathbf{0} \end{Bmatrix}$$

where

$$\begin{cases} \mathbf{K}^{\theta\theta} = & \int_l \left[\left(\mathbf{B}^{\theta}\right)^T EI \mathbf{B}^{\theta} \right] dx \\ \mathbf{K}^{SS} = & \int_l \left[(N_S)^T \frac{1}{\kappa GA} N_S \right] dx \\ \mathbf{K}^{\theta S} = & \int_l \left[\left(\mathbf{N}^{\theta}\right)^T N_S \right] dx \\ \mathbf{K}^{vS} = & -\int_l \left[(\mathbf{B}^{v})^T N_S \right] dx \end{cases}$$

Timoshenko beam: mixed approach & static condensation II

- Solve last equation wrt $\hat{S}$

$$\left(\mathbf{K}^{\theta S}\right)^T \hat{\boldsymbol{\theta}} + \left(\mathbf{K}^{vS}\right)^T \hat{\mathbf{v}} - \mathbf{K}^{SS}\hat{\mathbf{S}} = \mathbf{0}$$

- Therefore:

$$\hat{\mathbf{S}} = \left(\mathbf{K}^{SS}\right)^{-1} \left[\left(\mathbf{K}^{\theta S}\right)^T \hat{\boldsymbol{\theta}} + \left(\mathbf{K}^{vS}\right)^T \hat{\mathbf{v}}\right]$$

- Hence:

$$\begin{bmatrix} \widetilde{\mathbf{K}^{\theta\theta}} & \widetilde{\mathbf{K}^{\theta v}} \\ \left(\widetilde{\mathbf{K}^{\theta v}}\right)^T & \mathbf{K}^{vv} \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{\theta}} \\ \hat{\mathbf{v}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ .... \end{Bmatrix}$$

where

$$\begin{cases} \widetilde{\mathbf{K}^{\theta\theta}} = \mathbf{K}^{\theta\theta} + \left(\mathbf{K}^{SS}\right)^{-1}\left(\mathbf{K}^{\theta S}\right)^T \\ \widetilde{\mathbf{K}^{\theta v}} = \mathbf{K}^{\theta S}\left(\mathbf{K}^{SS}\right)^{-1}\left(\mathbf{K}^{vS}\right)^T \\ \widetilde{\mathbf{K}^{vv}} = \mathbf{K}^{vS}\left(\mathbf{K}^{SS}\right)^{-1}\left(\mathbf{K}^{vS}\right)^T \end{cases}$$

Timoshenko beam: enhanced approach I

- Start from a total potential energy approach (displacement based approach)

$$\Pi(v,\theta) = \frac{1}{2}\int_l \left[EI\chi^2\right] dx + \frac{1}{2}\int_l \left[\kappa GA\gamma^2\right] dx - \Pi^{ext}$$

- ◦ Enhance strain field (i.e., the one suffering locking) with a unknown field $\tilde{\gamma}$

$$\begin{cases} \chi(\theta) = \theta' \\ \gamma(v,\theta,\tilde{\gamma}) = v' - \theta + \tilde{\gamma} \end{cases}$$

- ◦ Hence

$$\Pi = \Pi(v,\theta,\tilde{\gamma})$$

- ⋆ Possible to take variation wrt $v,\theta,\tilde{\gamma}$

Timoshenko beam: enhanced approach II

- **Simplest choice for enhanced approach**

★ Piecewise linear (continuous) displacement/rotation fields:

- two node element
- two dofs per node: 1 transv. displ. + 1 rotation

$$\begin{cases} v = \mathbf{N}\hat{\mathbf{v}} \\ \theta = \mathbf{N}\hat{\boldsymbol{\theta}} \end{cases}$$

with

$$\mathbf{N} = \left[1 - \frac{x}{l} \quad , \quad \frac{x}{l}\right]$$

★ Linear enhanced shear strain field

$$\tilde{\gamma} = x\hat{\gamma}$$

- like having one internal node with one dof per node
- one dof per node

✓ Possible as usual to perform static condensation of internal dof

✓ Also other possible approaches are possibles (e.g., under-integration, assumed strain)

Further readings I

- O.C.Zienkiewicz and R.L.Taylor, "The Finite Element Method", Butterworth-Heinemann (2005), vol. 2
  chapter 10

- E. Onate, Structural Analysis with the Finite Element Method. Linear Statics: Volume 2: Beams, Plates and Shells, Springer (2013)
  chapters 1 & 2