Introduction to the Finite Element Method (2)

Petr Kabele

Czech Technical University in Prague
Faculty of Civil Engineering
Czech Republic

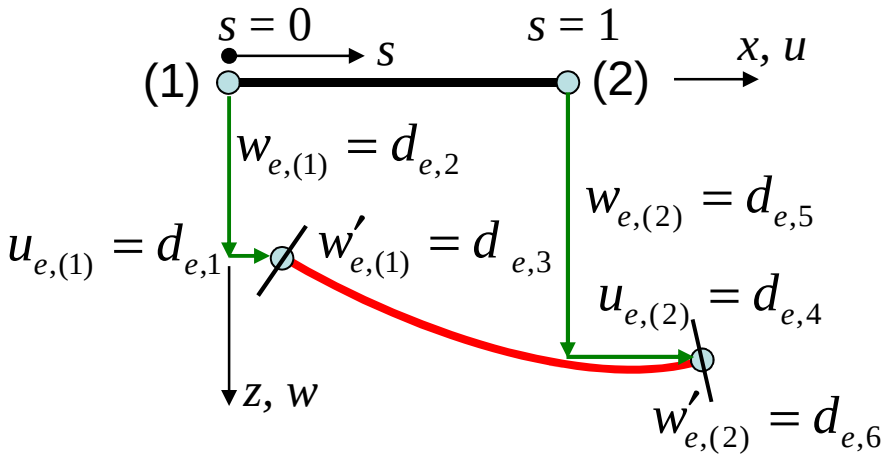
petr.kabele@fsv.cvut.cz ∗ people.fsv.cvut.cz/~pkabele

Outline

- Hermitian beam element
- Isoparametric 2-D continuum element
- Some types of commonly used elements
- Solution of equation systems
- Convergence of analysis results

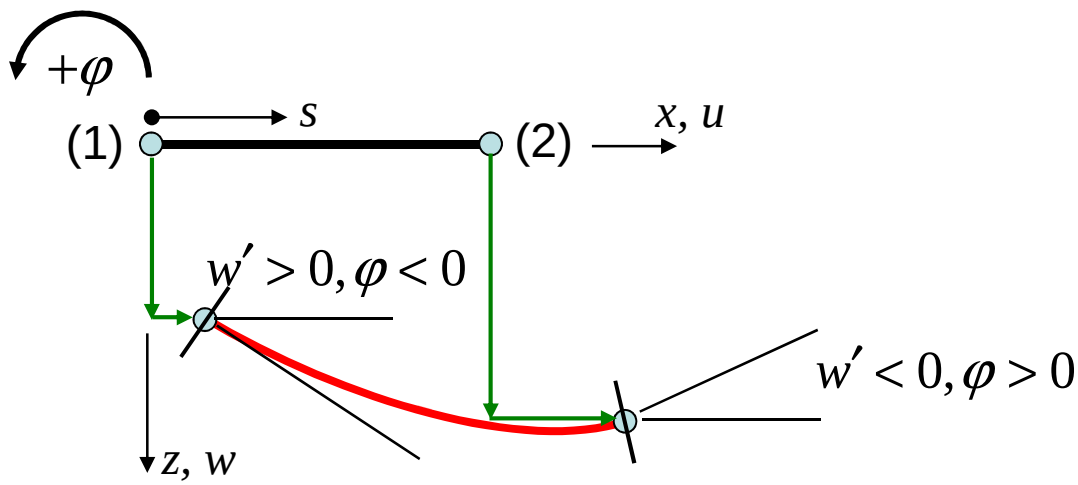
Hermitian beam element (in plane x-z)

Degrees of freedom



- The generalized strain κ associated with bending of beams involves the 2nd derivative of the field variable w (deflection). To construct conformable elements, C^1 continuity is required. This is achieved by considering deflections w and their derivatives $\dfrac{dw}{dx}$ as degrees of freedom (primary variables to be solved).
- The generalized strain ε associated with axial stretching involves the 1st derivative of axial displacement u. Thus C^0 continuity is sufficient.
- Considering 2-node element of length l, there are totally 6 degrees of freedom.

- Note that with the chosen degrees of freedom $\frac{dw}{dx} > 0$ corresponds to negative rotation $\varphi < 0$



- The approximated deflection within an element may be expressed as:

$$u(s) = N_{e,1}(s)\,d_{e,1} + N_{e,4}(s)\,d_{e,4}$$

$$w(s) = N_{e,2}(s)\,d_{e,2} + N_{e,3}(s)\,d_{e,3} + N_{e,5}(s)\,d_{e,5} + N_{e,6}(s)\,d_{e,6}$$

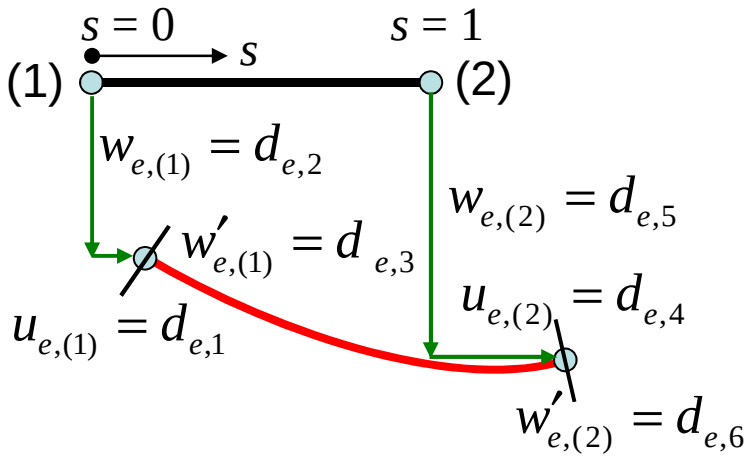
$$\begin{Bmatrix} u(s) \\ w(s) \end{Bmatrix} = \begin{bmatrix} N_{e,1}(s) & 0 & 0 & N_{e,4}(s) & 0 & 0 \\ 0 & N_{e,2}(s) & N_{e,3}(s) & 0 & N_{e,5}(s) & N_{e,6}(s) \end{bmatrix} \begin{Bmatrix} d_{e,1} \\ d_{e,2} \\ d_{e,3} \\ d_{e,4} \\ d_{e,5} \\ d_{e,6} \end{Bmatrix}$$

$$\mathbf{u}(s) = \mathbf{N}_e \mathbf{d}_e$$

Derivation of shape functions for bending

- Approximated deflection must be consistent with the nodal DOF's:

$$w(0) = w_{e,(1)} = d_{e,2}$$
$$w'(0) = w'_{e,(1)} = d_{e,3}$$
$$w(1) = w_{e,(2)} = d_{e,5}$$
$$w'(1) = w'_{e,(2)} = d_{e,6}$$



- Thus each shape function must satisfy 4 conditions

$$N_{e,2}(0) = 1,\ N_{e,3}(0) = 0,\ N_{e,5}(0) = 0,\ N_{e,6}(0) = 0$$
$$\frac{dN_{e,2}}{dx}(0) = 0,\ \frac{dN_{e,3}}{dx}(0) = 1,\ \frac{dN_{e,5}}{dx}(0) = 0,\ \frac{dN_{e,6}}{dx}(0) = 0$$
$$N_{e,2}(1) = 0,\ N_{e,3}(1) = 0,\ N_{e,5}(1) = 1,\ N_{e,6}(1) = 0 \qquad (*)$$
$$\frac{dN_{e,2}}{dx}(1) = 0,\ \frac{dN_{e,3}}{dx}(1) = 0,\ \frac{dN_{e,5}}{dx}(1) = 0,\ \frac{dN_{e,6}}{dx}(1) = 1$$

- Note that

$$s = \frac{x - x_{e,(1)}}{l_e} \rightarrow \quad \frac{d}{dx} = \frac{d}{ds}\frac{ds}{dx} = \frac{1}{l_e}\frac{d}{ds}$$

- To allow enough flexibility, each $N_{e,i}$ for bending is assumed in the form of 3th order polynomial

$$N_{e,i}(s) = A_i + B_i s + C_i s^2 + D_i s^3$$

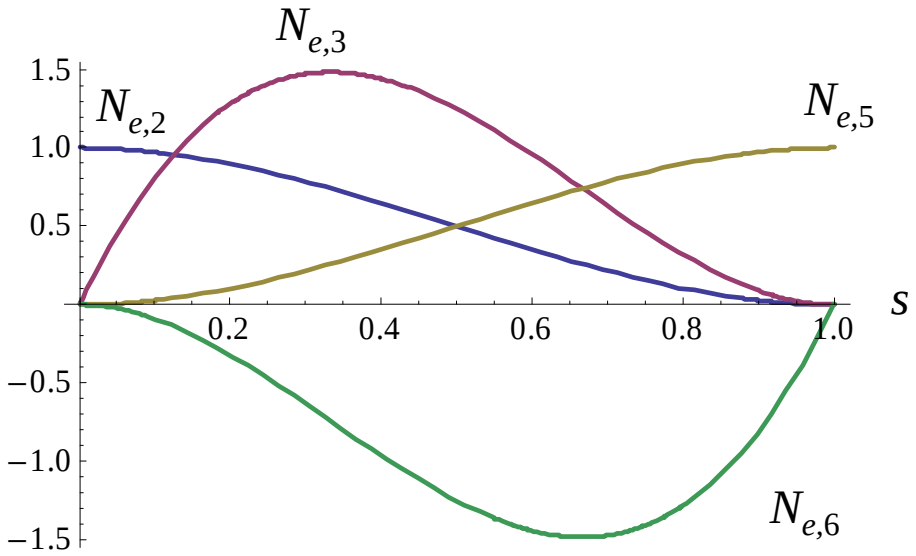
- The constants are solved from equations (*), which yield

$$N_{e,2}(s) = 1 - 3s^2 + 2s^3$$

$$N_{e,3}(s) = l_e\left(s - 2s^2 + s^3\right)$$

$$N_{e,5}(s) = 3s^2 - 2s^3$$

$$N_{e,6}(s) = l_e\left(-s^2 + s^3\right)$$



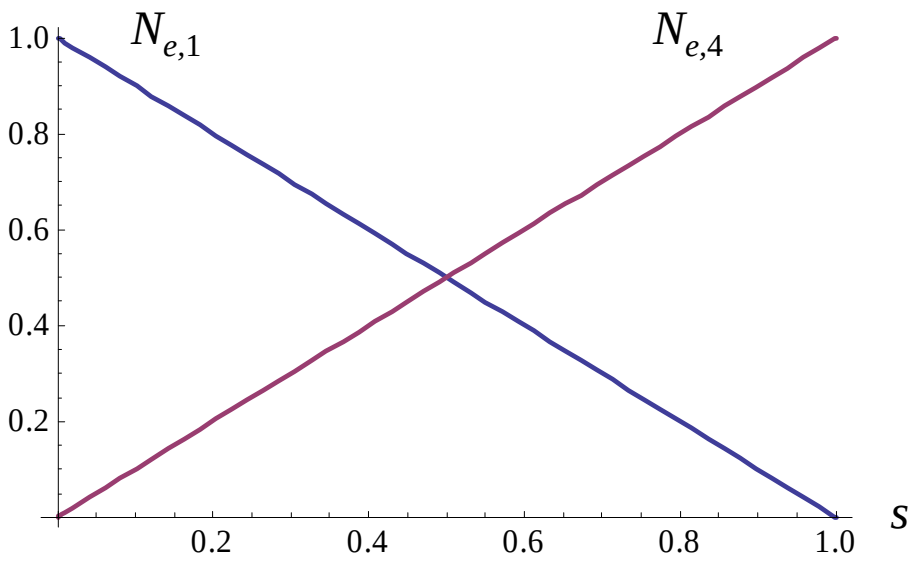
Shape functions for axial stretching

- For axial behavior, approximation by linear functions is sufficient for C^0 continuity. The shape functions must be consistent with the nodal displacements:

$$u(0) = u_{e,(1)} = d_{e,1}$$
$$u(1) = u_{e,(2)} = d_{e,4}$$

- The following functions satisfy these requirements:

$$N_{e,1}(s) = 1 - s$$
$$N_{e,4}(s) = s$$



Generalized strain

$$\begin{Bmatrix} \varepsilon \\ \kappa \end{Bmatrix} = \begin{Bmatrix} \dfrac{du}{dx} \\ -\dfrac{d^2 w}{dx^2} \end{Bmatrix}$$

but u and w are approximated as functions of $s = \dfrac{x - x_{e,(1)}}{l_e}$

So we use:

$$\frac{du}{dx} = \frac{du}{ds}\frac{ds}{dx} = \frac{1}{l_e}\frac{du}{ds}$$

$$\frac{d^2 w}{dx^2} = \frac{d}{dx}\left(\frac{dw}{dx}\right) = \frac{d}{ds}\frac{ds}{dx}\left(\frac{dw}{ds}\frac{ds}{dx}\right) = \left(\frac{ds}{dx}\right)^2 \frac{d^2 w}{ds^2} = \frac{1}{l_e^2}\frac{d^2 w}{ds^2}$$

$$\begin{Bmatrix} \varepsilon \\ \kappa \end{Bmatrix} = \begin{bmatrix} \frac{1}{l_e}\frac{dN_{e,1}}{ds} & 0 & 0 & \frac{1}{l_e}\frac{dN_{e,4}}{ds} & 0 & 0 \\ 0 & -\frac{1}{l_e^2}\frac{d^2N_{e,2}}{ds^2} & -\frac{1}{l_e^2}\frac{d^2N_{e,3}}{ds^2} & 0 & -\frac{1}{l_e^2}\frac{d^2N_{e,5}}{ds^2} & -\frac{1}{l_e^2}\frac{d^2N_{e,6}}{ds^2} \end{bmatrix} \begin{Bmatrix} d_{e,1} \\ d_{e,2} \\ d_{e,3} \\ d_{e,4} \\ d_{e,5} \\ d_{e,6} \end{Bmatrix}$$

$$= \mathbf{B}_e \mathbf{d}_e$$

```
(%i10) B(s):=''(ratsimp(matrix(
       [1/le*diff(N1(s),s,1),0,0,1/le*diff(N4(s),s,1),0,0],
       [0,-1/le^2*diff(N2(s),s,2),-1/le^2*diff(N3(s),s,2),
        0,-1/le^2*diff(N5(s),s,2),-1/le^2*diff(N6(s),s,2)]
       )));
```

(%o10) $B(s):=\begin{bmatrix} -\frac{1}{le} & 0 & 0 & \frac{1}{le} & 0 & 0 \\ 0 & -\frac{12\,s-6}{le^2} & -\frac{6\,s-4}{le} & 0 & \frac{12\,s-6}{le^2} & -\frac{6\,s-2}{le} \end{bmatrix}$

Generalized stress-generalized strain relation

$$\begin{Bmatrix} N \\ M \end{Bmatrix} = \begin{bmatrix} E_e A_e & 0 \\ 0 & E_e I_e \end{bmatrix} \begin{Bmatrix} \varepsilon \\ \kappa \end{Bmatrix}$$

Element stiffness matrix

$$\mathbf{K}_e = \int_{x=x_{e,(1)}}^{x_{e,(1)}+l_e} \mathbf{B}_e^T \mathbf{D}_e \mathbf{B}_e \, dx = \int_{s=0}^{1} \mathbf{B}_e^T \mathbf{D}_e \mathbf{B}_e \, l_e \, ds$$

Note:

$$s = \frac{x - x_{e,(1)}}{l_e} \rightarrow ds = \frac{dx}{l_e} \rightarrow dx = l_e ds$$

- In this case, the integration can be done explicitly

```
(%i14) Ke:integrate(transpose(B(s)).D.B(s)*le,s,0,1);
```

(%o14)

$$\begin{bmatrix} \frac{A\,E}{le} & 0 & 0 & -\frac{A\,E}{le} & 0 & 0 \\ 0 & \frac{12\,E\,I}{le^3} & \frac{6\,E\,I}{le^2} & 0 & -\frac{12\,E\,I}{le^3} & \frac{6\,E\,I}{le^2} \\ 0 & \frac{6\,E\,I}{le^2} & \frac{4\,E\,I}{le} & 0 & -\frac{6\,E\,I}{le^2} & \frac{2\,E\,I}{le} \\ -\frac{A\,E}{le} & 0 & 0 & \frac{A\,E}{le} & 0 & 0 \\ 0 & -\frac{12\,E\,I}{le^3} & -\frac{6\,E\,I}{le^2} & 0 & \frac{12\,E\,I}{le^3} & -\frac{6\,E\,I}{le^2} \\ 0 & \frac{6\,E\,I}{le^2} & \frac{2\,E\,I}{le} & 0 & -\frac{6\,E\,I}{le^2} & \frac{4\,E\,I}{le} \end{bmatrix}$$

Element vector of external forces

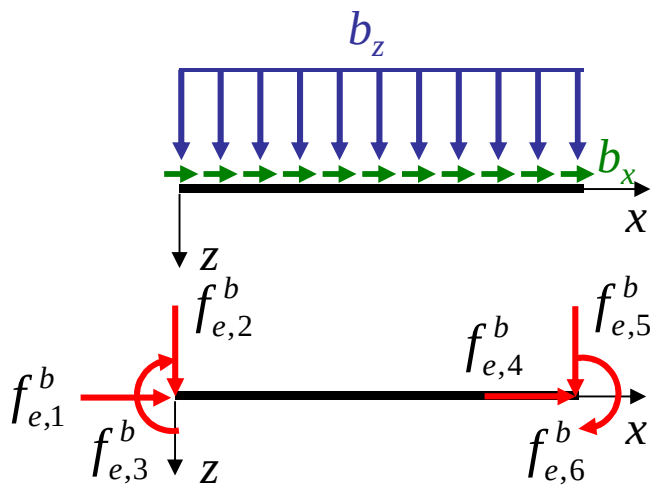
- Body force $\mathbf{b} = \begin{Bmatrix} b_x \\ b_z \end{Bmatrix}$ ... intensity per unit length (N/m)

$$\mathbf{f}_e^b = \int_{s=0}^{1} \mathbf{N}_e^T \mathbf{b}\, l_e ds$$

- For example for $\mathbf{b}$ constant:

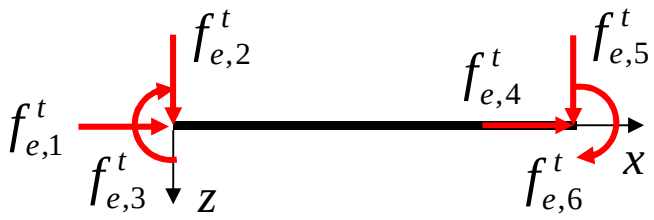
```
(%i15) fbe:integrate(transpose(N(s)).matrix([bx],[bz])*le,s,0,1);
```

(%o15)
$$\begin{bmatrix} \frac{bx\ le}{2} \\ \frac{bz\ le}{2} \\ \frac{bz\ le^2}{12} \\ \frac{bx\ le}{2} \\ \frac{bz\ le}{2} \\ -\frac{bz\ le^2}{12} \end{bmatrix}$$



Note that the positive orientation of forces (moments) corresponds to the positive orientation of DOF's

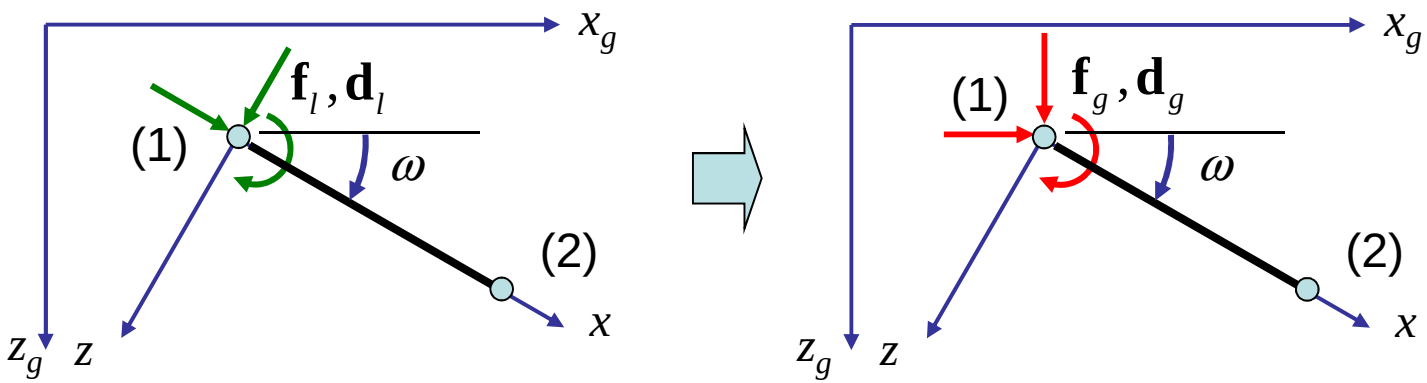
- Surface tractions → applied end forces (N) and moments (Nm)



Note that the positive orientation of forces (moments) corresponds to the positive orientation of DOF's

Transformation of coordinate system

- Up to now, all derivations were done assuming that the beam axis is collinear with the x-axis. We use axes x-z as element local system.
- When modeling a general structure, we usually refer all nodal degrees of freedom and nodal forces to a fixed global coordinate system x_g-z_g. Then some elements may be inclined with respect to the global axes and their stiffness matrices and load vectors must be transformed from the local system:



$$\cos\omega = \frac{x_{g(2)} - x_{g(1)}}{l_e}, \quad \sin\omega = \frac{z_{g(2)} - z_{g(1)}}{l_e}$$

- The vectors in global coordinates can be transformed to local by:

$$\mathbf{d}_l = \mathbf{T}\,\mathbf{d}_g$$
$$\mathbf{f}_l = \mathbf{T}\,\mathbf{f}_g$$

where $\mathbf{T} = \begin{bmatrix} \mathbf{T}_{sub} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_{sub} \end{bmatrix}$, $\quad \mathbf{T}_{sub} = \begin{bmatrix} \cos\omega & \sin\omega & 0 \\ -\sin\omega & \cos\omega & 0 \\ 0 & 0 & 1 \end{bmatrix}$

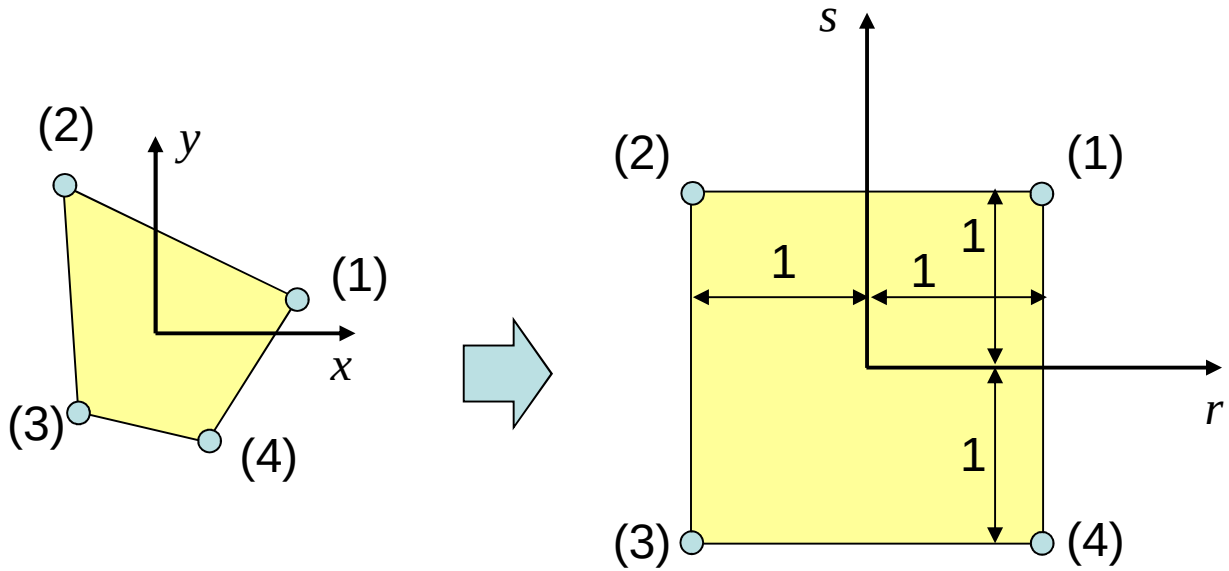
- Then

$$\mathbf{K}_l\,\mathbf{d}_l = \mathbf{f}_l$$
$$\mathbf{K}_l\,\mathbf{T}\mathbf{d}_g = \mathbf{T}\mathbf{f}_g$$
$$\mathbf{T}^T\,\mathbf{K}_l\,\mathbf{T}\mathbf{d}_g = \mathbf{f}_g$$

where we used the property $\mathbf{T}^{-1} = \mathbf{T}^T$

Isoparametric 2D continuum element

- Isoparametric elements use the same interpolation function for interpolation the element geometry and interpolation of its displacement field.
- Natural coordinates are introduced, which map the element into a square with side length equal to 2. For example:



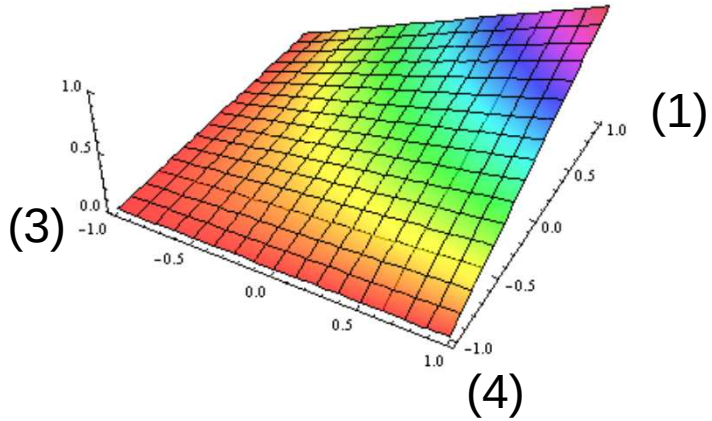
- For a 4-node quadrilateral element

$$\begin{Bmatrix} x(r,s) \\ y(r,s) \end{Bmatrix} = \begin{bmatrix} N_{(1)} & 0 & N_{(2)} & 0 & N_{(3)} & 0 & N_{(4)} & 0 \\ 0 & N_{(1)} & 0 & N_{(2)} & 0 & N_{(3)} & 0 & N_{(4)} \end{bmatrix} \begin{Bmatrix} x_{e,(1)} \\ y_{e,(1)} \\ x_{e,(2)} \\ y_{e,(2)} \\ x_{e,(3)} \\ y_{e,(3)} \\ x_{e,(4)} \\ y_{e,(4)} \end{Bmatrix}$$

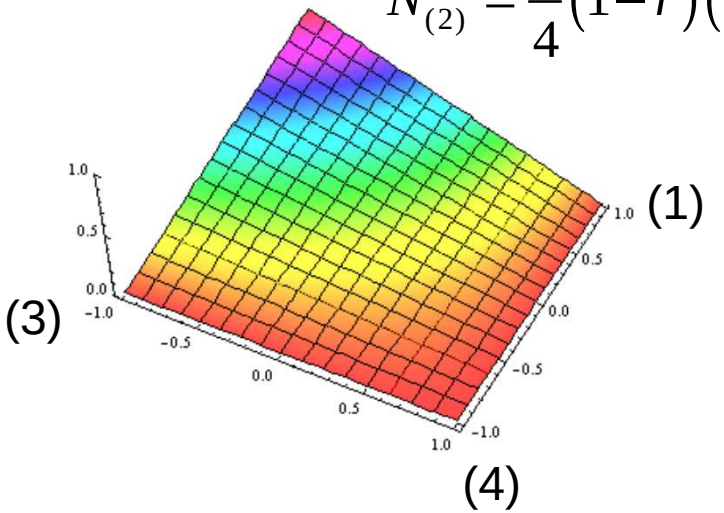
$$\mathbf{x} = \mathbf{N}_e \mathbf{x}_e$$

where

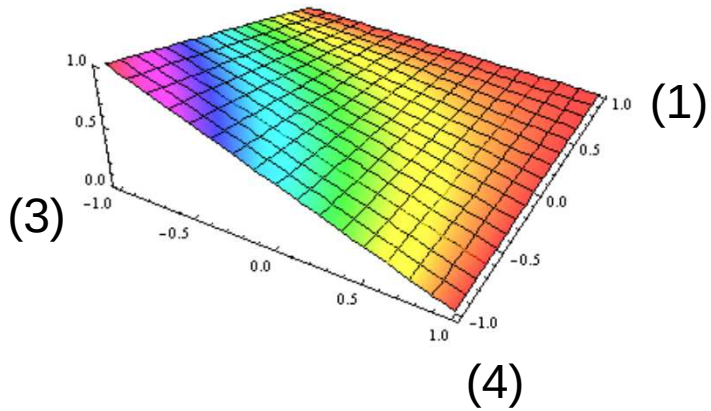
$$N_{(1)} = \frac{1}{4}(1+r)(1+s)$$



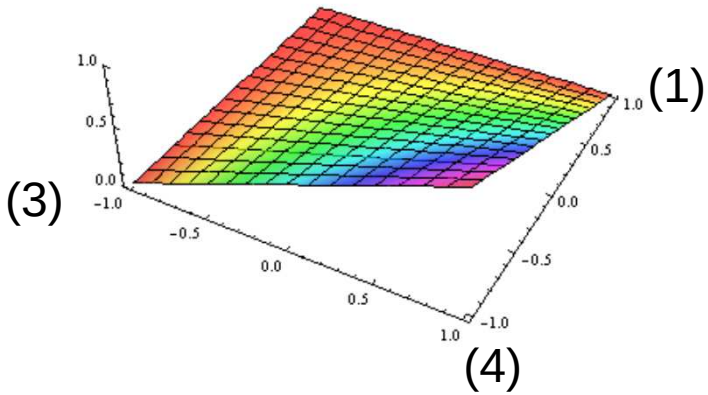
$$N_{(2)} = \frac{1}{4}(1-r)(1+s)$$



$$N_{(3)} = \frac{1}{4}(1-r)(1-s)$$



$$N_{(4)} = \frac{1}{4}(1+r)(1-s)$$



- Isoparametric elements are often used in analysis of 2D continuum problems (plane stress, plane strain, axial symmetry).
- In these problems, strain contains the first derivatives of displacement. C^0 continuity is therefore sufficient. Nodal displacements $u_{e,(i)}$, $v_{e,(i)}$ are used as the degrees of freedom. Then displacement within an element is approximated by

$$\begin{Bmatrix} u(r,s) \\ v(r,s) \end{Bmatrix} = \begin{bmatrix} N_{(1)} & 0 & N_{(2)} & 0 & N_{(3)} & 0 & N_{(4)} & 0 \\ 0 & N_{(1)} & 0 & N_{(2)} & 0 & N_{(3)} & 0 & N_{(4)} \end{bmatrix} \begin{Bmatrix} u_{e,(1)} \\ v_{e,(1)} \\ u_{e,(2)} \\ v_{e,(2)} \\ u_{e,(3)} \\ v_{e,(3)} \\ u_{e,(4)} \\ v_{e,(4)} \end{Bmatrix}$$

$$\mathbf{u} = \mathbf{N}_e \mathbf{d}_e$$

- Recall that when constructing the element stiffness matrices and force vectors, the derivatives of approximated u and v with respect to x, y are necessary. However, the spatial variation of displacements is expressed through the shape functions which depend on the natural coordinates r, s . Also note, that we can express x, y in terms of r, s through $\mathbf{x} = \mathbf{N}_e \mathbf{x}_e$ but the inverse mapping is generally not available.
- To obtain the desired derivatives we first use the chain rule as follows:

$$\begin{Bmatrix} \dfrac{\partial}{\partial r} \\ \dfrac{\partial}{\partial s} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial x}{\partial r} & \dfrac{\partial y}{\partial r} \\ \dfrac{\partial x}{\partial s} & \dfrac{\partial y}{\partial s} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} \end{Bmatrix}$$

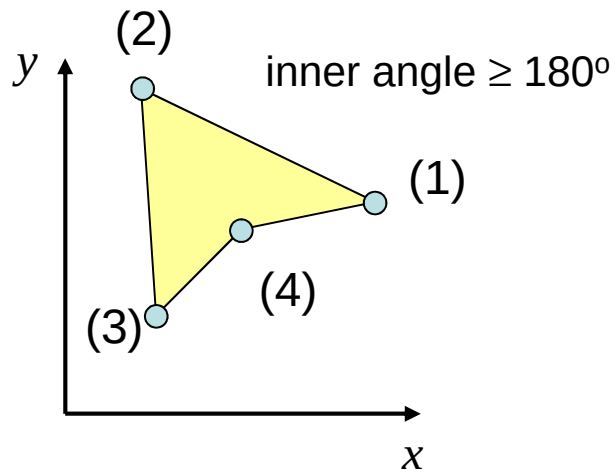
$$\frac{\partial}{\partial \mathbf{r}} = \mathbf{J} \frac{\partial}{\partial \mathbf{x}}$$

$\mathbf{J}$ ... Jacobian operator or Jacobian matrix

- Then, provided the inverse of $\mathbf{J}$ exists:

$$\begin{Bmatrix} \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} \end{Bmatrix} = \begin{bmatrix} \left(\mathbf{J}^{-1}\right)_{1,1} & \left(\mathbf{J}^{-1}\right)_{1,2} \\ \left(\mathbf{J}^{-1}\right)_{2,1} & \left(\mathbf{J}^{-1}\right)_{2,2} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial}{\partial r} \\ \dfrac{\partial}{\partial s} \end{Bmatrix} \qquad \text{or} \qquad \frac{\partial}{\partial \mathbf{x}} = \mathbf{J}^{-1} \frac{\partial}{\partial \mathbf{r}}$$

Note: the inverse does not exist if there is not unique relationship between r, s and x, y. This may happen, for example if the element is overly distorted.



- Approximated strain distribution within the element is obtained as

$$\begin{Bmatrix} \varepsilon_x(r,s) \\ \varepsilon_y(r,s) \\ \gamma_{xy}(r,s) \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} u(r,s) \\ v(r,s) \end{Bmatrix}$$

matrix $\mathbf{B}_e$

$$\boxed{= \begin{bmatrix} \left(\left(\mathbf{J}^{-1}\right)_{1,1} \dfrac{\partial}{\partial r} + \left(\mathbf{J}^{-1}\right)_{1,2} \dfrac{\partial}{\partial s} \right) & 0 \\ 0 & \left(\left(\mathbf{J}^{-1}\right)_{2,1} \dfrac{\partial}{\partial r} + \left(\mathbf{J}^{-1}\right)_{2,2} \dfrac{\partial}{\partial s} \right) \\ \left(\left(\mathbf{J}^{-1}\right)_{2,1} \dfrac{\partial}{\partial r} + \left(\mathbf{J}^{-1}\right)_{2,2} \dfrac{\partial}{\partial s} \right) & \left(\left(\mathbf{J}^{-1}\right)_{1,1} \dfrac{\partial}{\partial r} + \left(\mathbf{J}^{-1}\right)_{1,2} \dfrac{\partial}{\partial s} \right) \end{bmatrix} \begin{bmatrix} N_{(1)} & 0 & N_{(2)} & 0 & N_{(3)} & 0 & N_{(4)} & 0 \\ 0 & N_{(1)} & 0 & N_{(2)} & 0 & N_{(3)} & 0 & N_{(4)} \end{bmatrix}} \begin{Bmatrix} u_{e,(1)} \\ v_{e,(1)} \\ u_{e,(2)} \\ v_{e,(2)} \\ u_{e,(3)} \\ v_{e,(3)} \\ u_{e,(4)} \\ v_{e,(4)} \end{Bmatrix}$$

$$\boldsymbol{\varepsilon} = \mathbf{B}_e \, \mathbf{d}_e$$

- The element stiffness matrix is obtained by integration over the volume of the element

$$\mathbf{K}_e = \int_A \mathbf{B}_e^T \mathbf{D} \mathbf{B}_e \; t \, dxdy$$

$\mathbf{D}$ ... constitutive matrix (stress-strain relation)

t ... element thickness

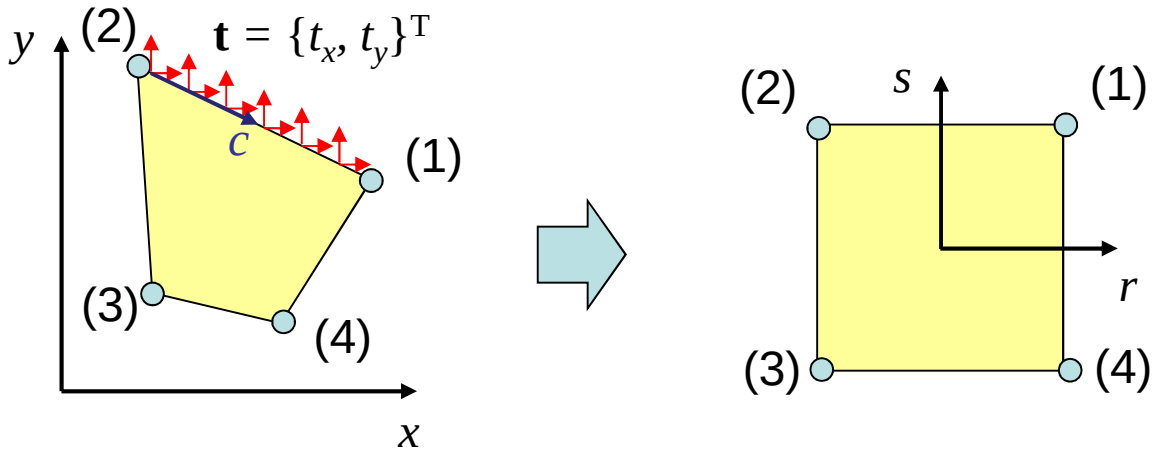
- It can be shown that $dx\,dy = \det \mathbf{J} \, dr \, ds$ and thus

$$\mathbf{K}_e = \int_{r=-1}^{1} \int_{s=-1}^{1} \mathbf{B}_e^T \mathbf{D} \mathbf{B}_e \; t \; \det \mathbf{J} \, dr \, ds$$

- Similarly, the nodal forces vector associated with body forces is

$$\mathbf{f}_e^b = \int_{r=-1}^{1} \int_{s=-1}^{1} \mathbf{N}_e^T \mathbf{b} \; t \; \det \mathbf{J} \, dr \, ds$$

- To calculate the nodal forces vector associated with surface tractions, integration must be performed along the loaded element boundary, e.g.:



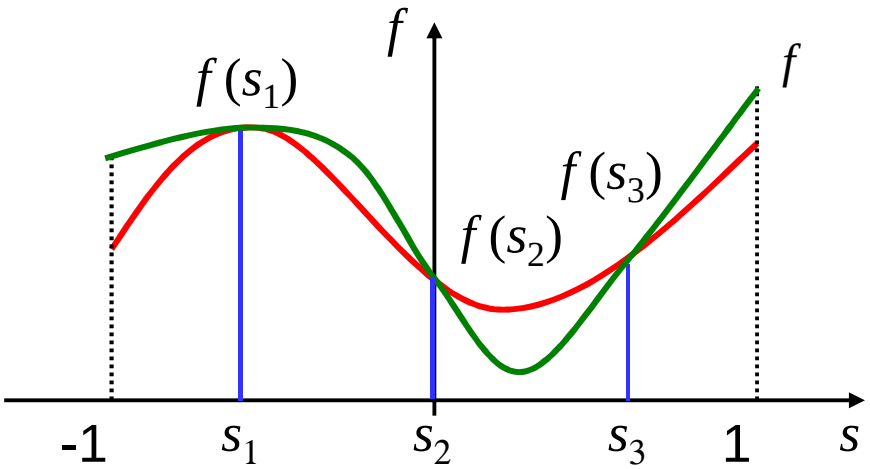
$$dc = \det \mathbf{J}_c \, dr$$

$$\mathbf{f}_e^t = \int_c \mathbf{N}^T \mathbf{t}\, t \, dc \quad \Rightarrow \quad \det \mathbf{J}_c = \sqrt{\left(\frac{\partial x}{\partial r}\right)^2 + \left(\frac{\partial y}{\partial r}\right)^2} \quad \Rightarrow \quad \mathbf{f}_e^t = \int_{r=-1}^{1} \mathbf{N}^T \mathbf{t}\, t \det \mathbf{J}_c \, dr$$

$$\frac{\partial x}{\partial r} = \frac{x_{(1)} - x_{(2)}}{2}, \quad \frac{\partial y}{\partial r} = \frac{y_{(1)} - y_{(2)}}{2}$$

Idea of Gauss numerical integration

- The calculation of element stiffness matrix and nodal force vectors involves integration over the element volume or contour. Generally, it is not efficient (or even possible) to evaluate these integrals explicitly; instead numerical integration is used.
- The principle of Gauss numerical integration consist in approximating the integrand with a polynomial at a given number of points and integrating this polynomial.



- The sought n be then approximately expressed as:

$$\int_{-1}^{1} f(s)\, ds \cong \sum_{i=1}^{n} w_i f(s_i)$$

n … number of integration points

w_i … weight of integration point i

s_i … coordinate of integration point i

n	s_i	w_i
1	0	2
2	$\pm 1/\sqrt{3}$	1
3	0	$8/9$
	$\pm\sqrt{3/5}$	$5/9$
4	$\pm\sqrt{\left(3-2\sqrt{6/5}\right)/7}$	$\frac{18+\sqrt{30}}{36}$
	$\pm\sqrt{\left(3+2\sqrt{6/5}\right)/7}$	$\frac{18-\sqrt{30}}{36}$

www.wikipedia.org

- By using n integration points, polynomials of order up to at most $(2n\text{-}1)$ are integrated exactly by the above formula.

- In multiple dimensions, the formula for one dimension can be used successively:

$$\int_{r=-1}^{1}\int_{s=-1}^{1} f(r,s)\,dr\,ds \cong \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j f\left(r_i, s_j\right)$$

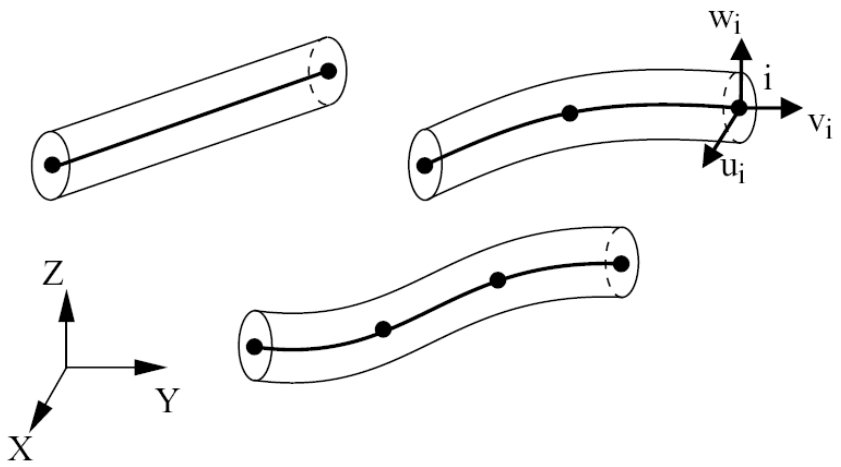
$$\int_{r=-1}^{1}\int_{s=-1}^{1}\int_{t=-1}^{1} f(r,s,t)\,dr\,ds\,dt \cong \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} w_i w_j w_k f\left(r_i, s_j, t_k\right)$$

Some types of commonly used elements

- *The review in this section is based on some of elements that are implemented in general purpose program ADINA®. They are common types of elements that are available in most FEM programs. However, each program may use different conventions for, e.g. direction, orientation and notation of degrees of freedom and nodal forces, input of loads, plotting of results, etc. The implementation of individual elements, such as interpolation functions etc. may also differ in some cases.*

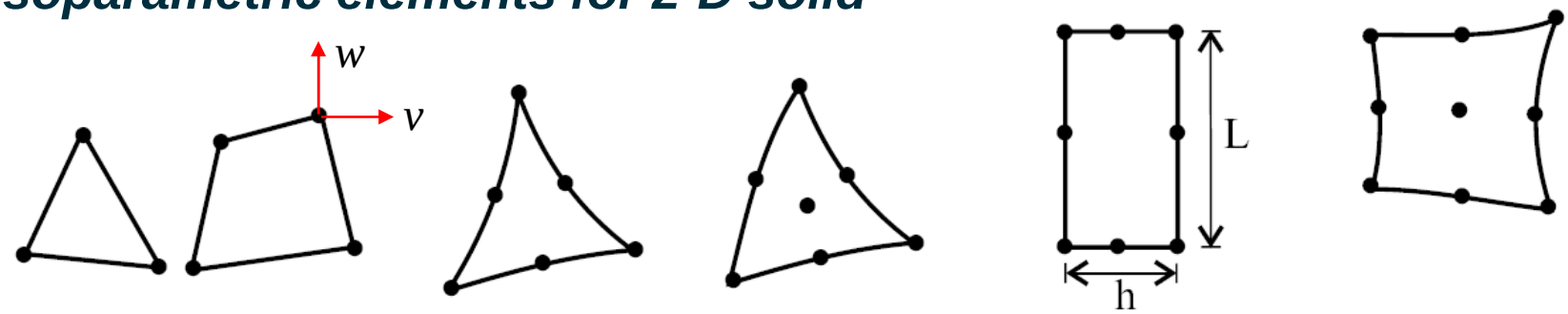
- A finite element type is defined by:
  - underlying theory (beam, plate, solid, ...)
  - geometrical dimension (linear, planar, spatial, …)
  - number and arrangement of nodes (shape)
  - used interpolation functions
  - number and type of degrees of freedom in each node
  - e.g. in Example 2 we used B-E beam, 1-dimensional linear element with 2 nodes, cubic interpolation functions, and 2 DOF's per node.

Truss and cable elements

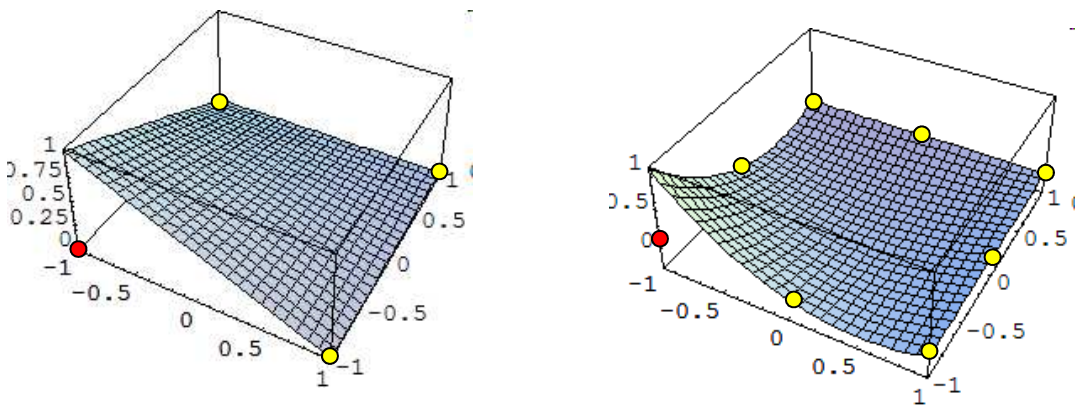


- number of nodes (shape): 2 (line), 3, 4 (curve)
- curved element transmits only axial force – behaves like cable
- global DOF's: translations u, v, w per node in 3-D (can be used also for 2-D, 1-D)
- number of integration points: 1~4 along element length (1 sufficient for truss)
- element output: axial strain, stress, force in integration points and others
- can be used to model truss structures, cable structures, springs, reinforcement in concrete (compatibility with solid elements)

Isoparametric elements for 2-D solid



- number of nodes (shape): 3~9 (generalized quadrilateral, generalized triangle)
- global DOF's: 2 per node - translations v, w in plane y-z (!)
- interpolation functions: bilinear to quadratic, depending on the number of nodes, e.g.

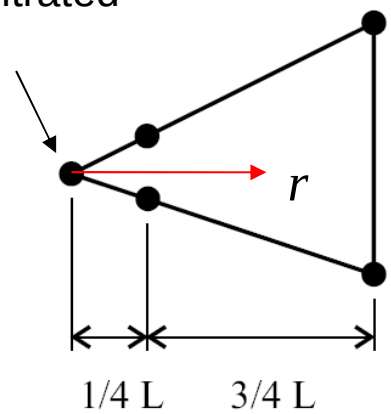


- Triangular elements may be created by collapsing quadrilaterals (nodes on one side of element are concentrated in to one and interpolation functions are modified)

constant strain

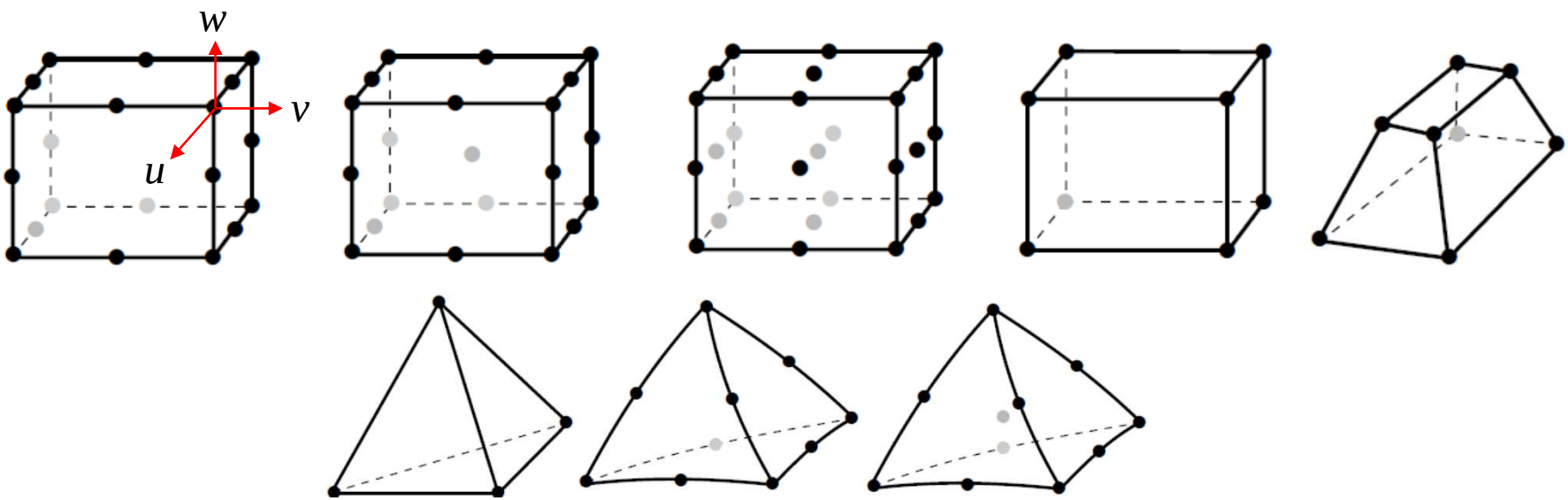
- Collapsing the element without modifying the interpolation functions leads to element for linear elastic fracture analysis with $\frac{1}{\sqrt{r}}$ singularity in the vicinity of a crack tip

3 nodes concentrated into 1



- number of integration points:
  - quadrilaterals 2×2 to 6×6 (typically 2×2 for quadrilaterals, 3×3 for others)
  - triangles 1 to 13
- element output: strain, stress, plastic strain, yield function and others in integration points
- recommended use:
  - plane stress, plane strain, axially symmetric problems
  - 9 node quadrilateral most effective
  - 8, 9 node elements most efficient as rectangular elements with aspect ratio > 1:10
  - 3, 4 node elements not effective when bending effect is significant (e.g. beam bending analyzed as 2-D solid)

Isoparametric elements for 3-D solid



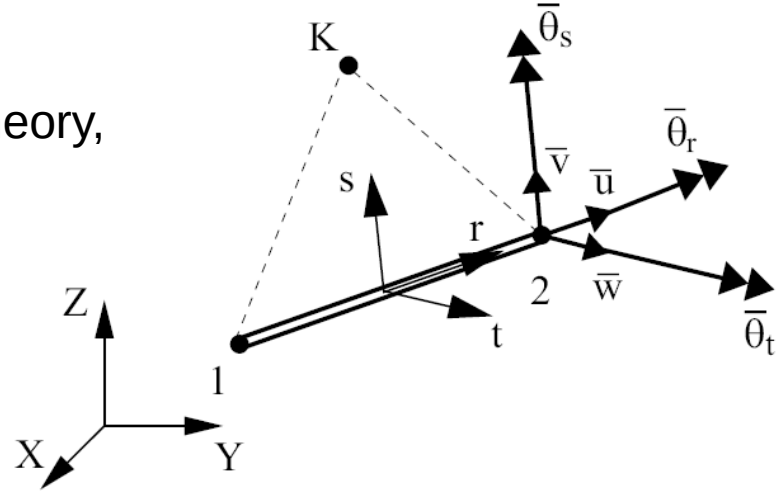
- number of nodes (shape): 4~27 (tetrahedron ~ brick)
- global DOF's: 3 per node - translations u, v, w

- number of integration points:
  - $2\times2\times2$ to $6\times6\times6$ (typically $2\times2\times2$ for 8-node, $3\times3\times3$ for others)

- element output: strain, stress, plastic strain, yield function and others in integration points

- recommended use:
  - problems where description of 3-D stress state is inevitable
  - 27 node most accurate but most costly
  - 20 node usually most effective
  - 20 node elements most efficient as rectangular
  - 8-node 3, 4 node elements not effective when bending effect is significant (e.g. beam bending analyzed as 2-D solid)

Hermitian beam element

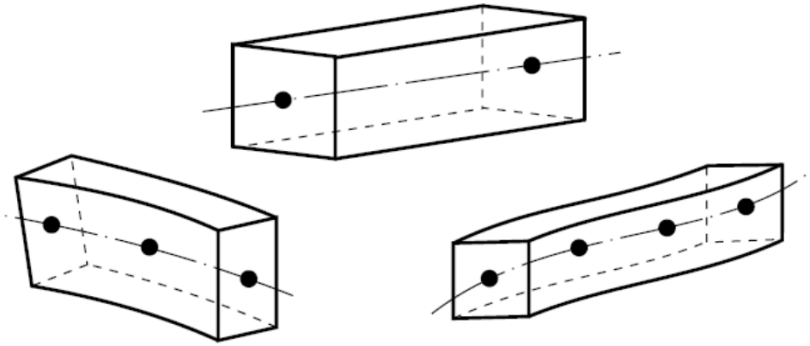


- beam element based on Bernoulli-Euler theory, possibly corrected for shear effects
- application for 2-D, 3-D
- number of nodes (shape): 2 (straight line, constant cross-section)
- degrees of freedom per node: 6
  - 3 translations
  - 3 rotations
  - in 2-D use only relevant 2 translations 1 rotation
- interpolation functions: cubic for transversal translations (bending), linear for axial translation and torsion

- possible input:
  - shape and dimensions of cross-section (or cross-sectional moduli) and material model (possibly nonlinear) or
  - moment-curvature and axial force-axial strain relationship (possibly nonlinear and coupled)

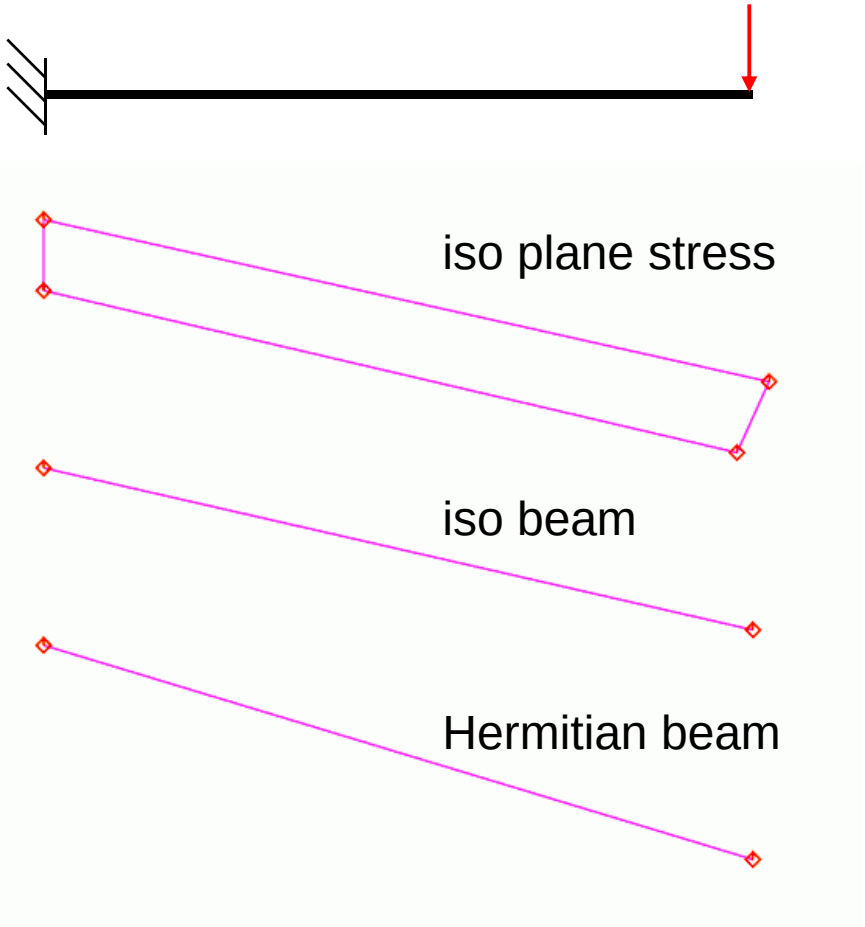
Isoparametric beam element

- isoparametric beam element based on Timoshenko theory
- application for 2-D, 3-D
- number of nodes (shape): 2 (straight line), 3~4 (curved in plane)
- only rectangular cross-section
- degrees of freedom per node: 6
  - 3 translations
  - 3 rotations
  - in 2-D use only relevant 2 translations 1 rotation



- 2 node element and 3~4 node element if nodes are not regularly spaced typically suffers shear locking (shear deformations are not represented with sufficient accuracy) → very stiff behavior, fine discretization required
- recommended use: curved geometry, large displacements (otherwise Hermitian element performs better)

- Example: locking of isoparametric elements

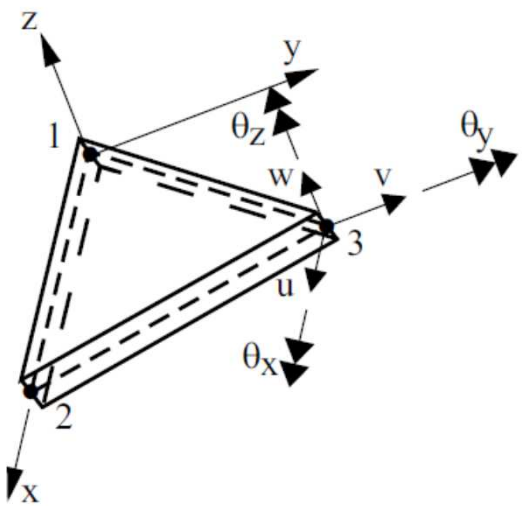


w = -3.02408

w = -3.02400

w = -4.00000 ... exact

Plate elements



- number of nodes (shape): 3 (triangle)
- DOF's: 6 per node – 3 translations and 3 rotations
- superposition of membrane and bending parts
- membrane part: 3-node constant strain element, plane stress
- bending part: element based on Kirchhoff theory of thin plates
- the element does not model shear deformations (Kirchhoff assumption)
- the element does not show locking
- suitable for modeling of very thin plates and shells
- possible outputs: stress resultants at integration points, nodal forces

Solution of equation systems

From the previous derivation of FEM, we can generalize the following facts:

- The governing equations and boundary conditions for elastic problems lead to a system of linear algebraic equations in the form of

$$\mathbf{K}\mathbf{d} = \mathbf{f}$$

  where $\mathbf{K}$ = global stiffness matrix, $\mathbf{d}$ = global vector of nodal unknowns (displacements, rotations), $\mathbf{f}$ = global vector of external nodal loads (forces, moments).

- The number of equations corresponds to the number of free (unsupported) degrees of freedom (usually many).
- The matrix $\mathbf{K}$ is usually sparse (many zero elements); reordering the vectors $\mathbf{d}$ and $\mathbf{f}$ it can be rearranged to become banded (nonzero elements concentrated around the diagonal).

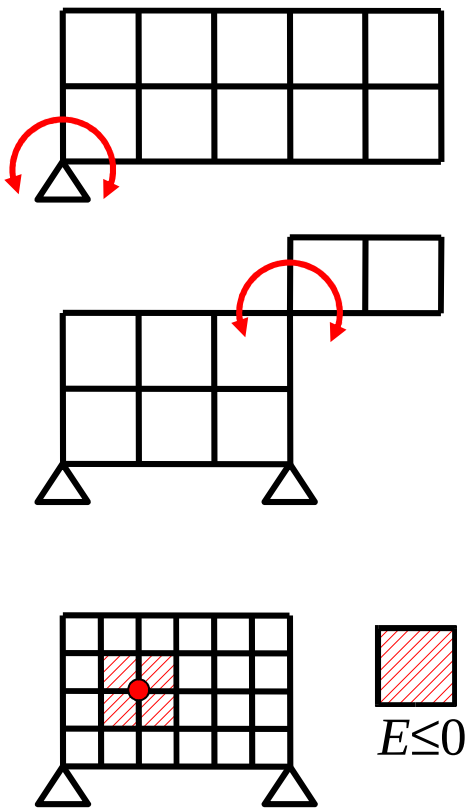
- In a well posed problem, the matrix $\mathbf{K}$ is positive definite:

$$\mathbf{d}^T \mathbf{K} \mathbf{d} > 0 \quad \forall\, \mathbf{d}$$

$$\det \mathbf{K} > 0$$

$\mathbf{K}$ is not positive definite e.g. in the following cases:

- the structure or its part in not sufficiently constrained to prevent rigid body motion
- the structural model contains degrees of freedom with zero or negative stiffness
- the material in a portion of the structure became unstable (e.g. as a result of softening, damage – will be discussed later).



- $\mathbf{K}$ is usually symmetric; it may not be symmetric e.g. when certain material models are used (non-associative plasticity, friction, etc.)

Basically, there exist two classes of methods to solve the type of equation systems arising from FEM:

- direct solves
- iterative solvers

Direct solution methods

- algorithm performs a number of steps and operations the are exactly predetermined by the number of equations and properties of the system matrix
- algorithm based on Gauss elimination – slow, high memory demands
- sparse solvers – robust and reliable, less memory demanding, 2 orders faster than Gauss elimination

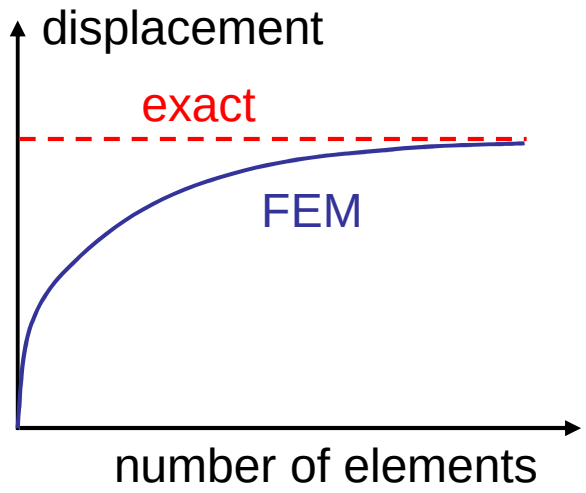
Iterative solution methods

- iterative and multigrid solvers - the solution is obtained iteratively, in a number of iterations which is not predetermined
- suitable for large problems, where the hardware memory capacity is not sufficient to hold the entire system of equations
- approximate solution of the system of equations is found iteratively by minimizing the norm of the difference between the LHS and RHS beyond given tolerance
- may not recognize system matrix which is not positive definite, problematic use for ill-conditioned matrices

Convergence of FE analysis results

- The finite element method is approximate numerical procedure for solving boundary value problems.
- Monotonic convergence – the approximate solution continuously approaches the exact mathematical solution of the BVP as the discretization is refined (number of elements increased).

- To assure monotonic convergence the elements must be **conformable** (see the criteria discussed in previous lecture, completeness and compatibility of displacement approximation).



- The rate of convergence depends on the element size, order of approximation polynomial, and material properties

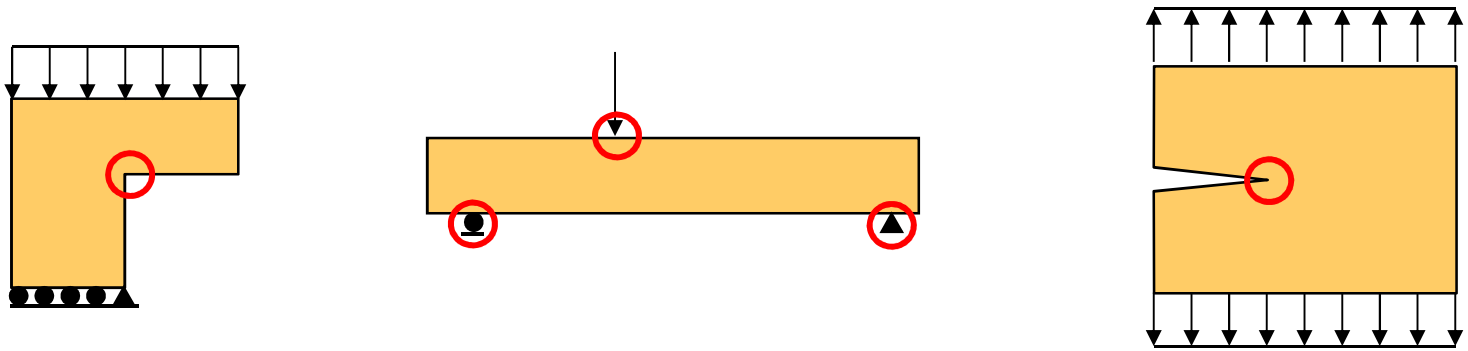
$$\left\|\mathbf{u}-\mathbf{u}_h\right\|_1 \leq c\, h^p \left\|\mathbf{u}\right\|_{p+1}$$

$\mathbf{u}$ ... exact solution
$\mathbf{u}_h$ ... approximate solution by FEM
h ... typical element size
p ... order of complete polynomial of approximation
c ... constant independent of h but dependent on material properties
$\left\|...\right\|_p$ ... Sobolev norm of order p, e.g.

$$\left(\left\|\mathbf{u}\right\|_0\right)^2 = \int_V \left(\sum_{i=1}^{3} (u_i)^2\right) dV \qquad \left(\left\|\mathbf{u}\right\|_1\right)^2 = \left(\left\|\mathbf{u}\right\|_0\right)^2 + \int_V \left(\sum_{i=1}^{3}\sum_{j=1}^{3}\left(\frac{\partial u_i}{\partial x_j}\right)^2\right) dV$$

- The above inequality implies, that the accuracy of the approximate solution by FEM can be improved by:
  - reducing the element size h ... h-convergence
  - increasing the approximation polynomial order p ... p-convergence

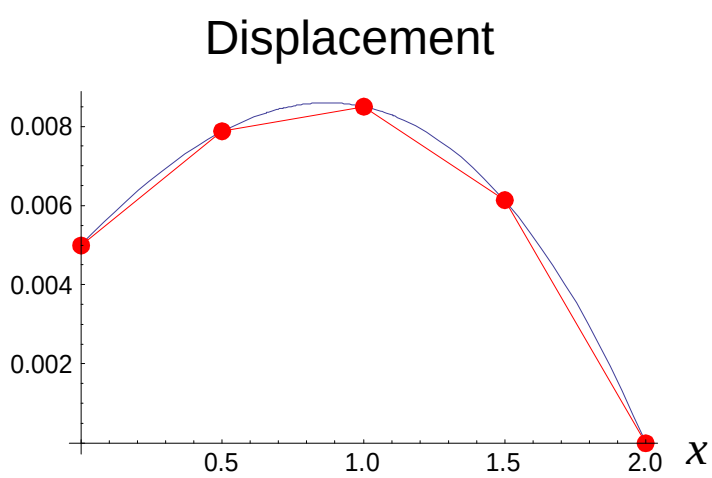
- Due to imposing constraint on the displacement field by approximation, the displacement-based FEM gives stiffer results (smaller displacements) than the exact solution.
- If the problem involves stress or strain singularities (esp. in 2D/3D continuum models: concentrated forces, point supports, sharp corners, crack tips, etc.), convergence in stress or strain at such locations cannot be obtained, unless special singularity elements are used. Overall convergence of displacements is slower. E.g.:

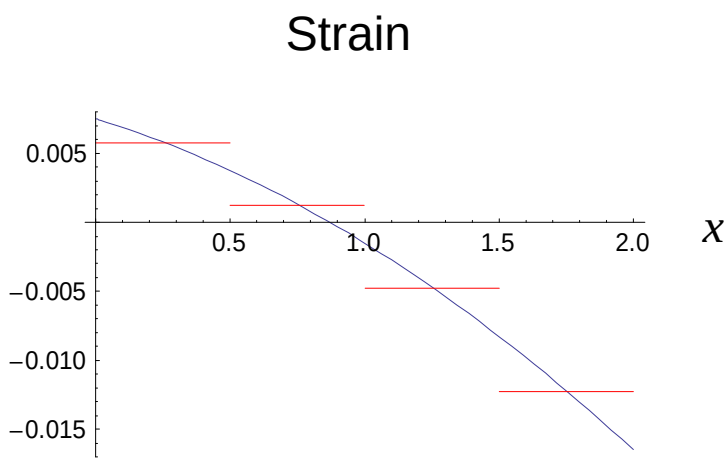


- Convergence discussed here refers to the convergence associated with the finite element discretization, not with nonlinear material behavior.

Superconvergence and optimal sampling points

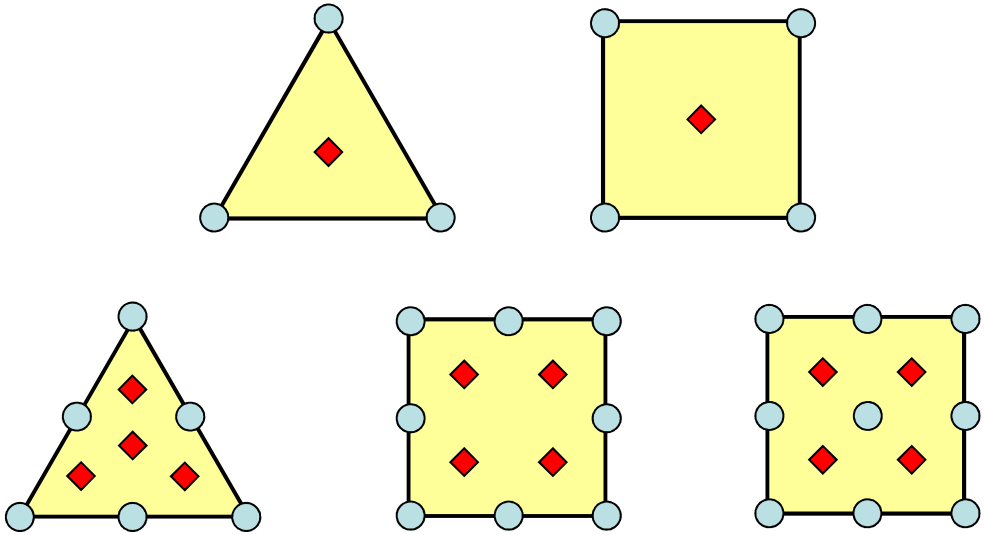
- Recall the FE solution of uniaxially stressed bar:





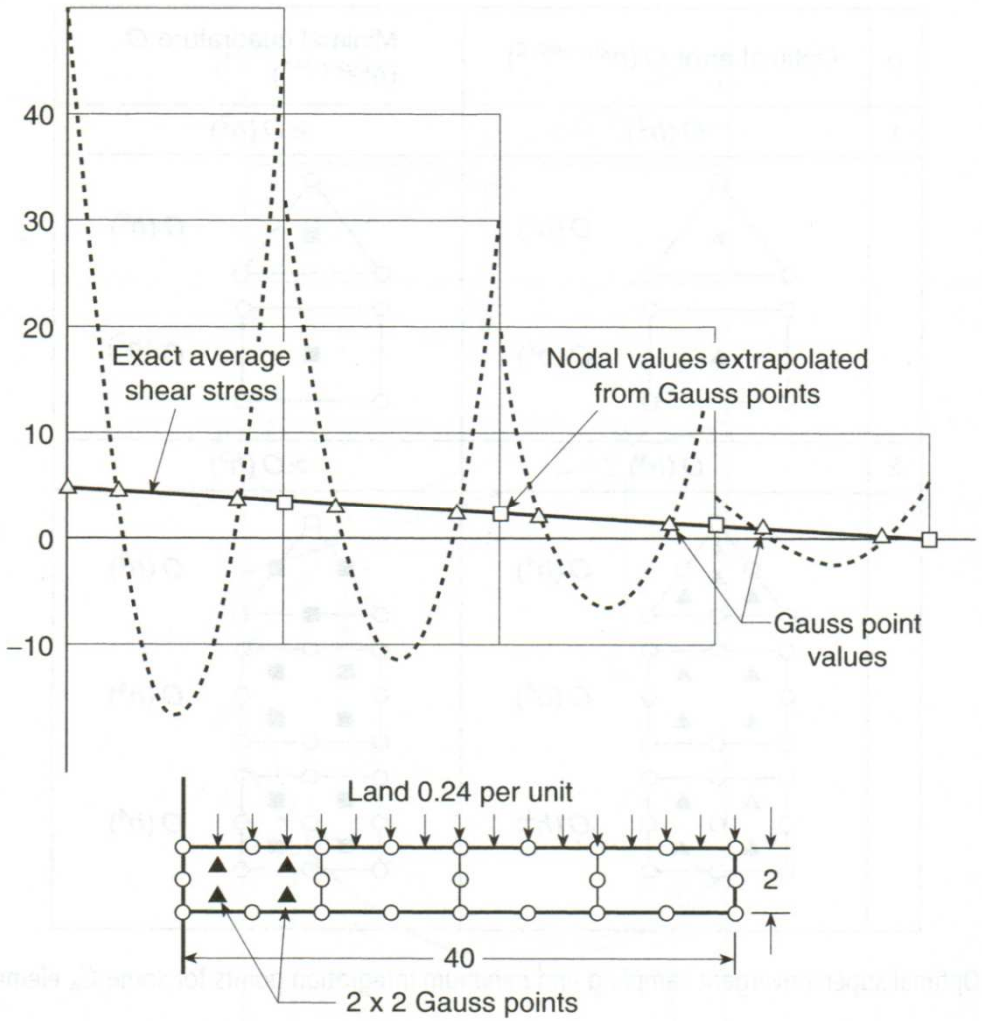
- Displacements (the quantity of primary approximation in finite elements) attain the best accuracy at FE nodes.
- Strain and stress (involving gradients of the primary approximation) attain the best accuracy at points within elements; their accuracy at nodes is poor.
- Order of convergence at such optimum points is one order higher than anticipated from the approximation polynomial ... supeconvergent points

- Optimal sampling points for strain and stress coincide with Gauss integration points



- Notes:
  - above applies to displacement-based FEM for elasticity problems
  - full superconvergence is not achieved with triangular elements or when elements are distorted (quadrilaterals or triangles), but the above points still provide superior results.

Example (Zienkiewicz & Taylor, 2000):



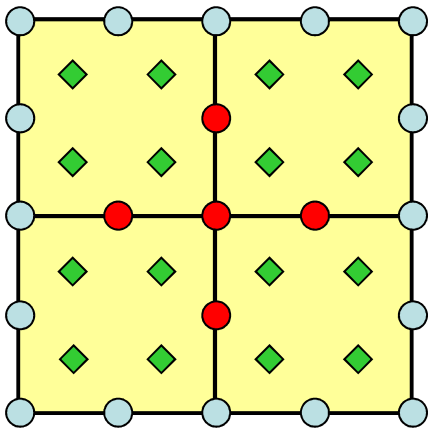
Recovery of strains and stresses

- Various methods may be used to recover smooth distributions of strain and stress.
- To this end, strain or stress may be approximated within elements using the same shape functions as those used for primary displacement approximation:

$$\boldsymbol{\varepsilon}(\mathbf{x}) \approx \mathbf{N}(\mathbf{x})\boldsymbol{\varepsilon}_{nodal}$$
$$\boldsymbol{\sigma}(\mathbf{x}) \approx \mathbf{N}(\mathbf{x})\boldsymbol{\sigma}_{nodal} \quad \text{*)}$$

- The nodal values of strain or stress may be obtained e.g. by:
  - **averaging** of strains/stresses evaluated at optimum sampling points of elements sharing a given node ... not the best for triangles and inadequate for higher order elements.
  - **L2 projection**: approximations *) are least-square fitted to the strains/stresses evaluated at optimum sampling points ... numerically costly.

- **superconvergent patch recovery** (SPR): direct polynomial smoothing of superconvergent values over a patch of several elements
  - stress is approximated by a polynomial over a patch of elements
  - coefficients of the polynomial are obtained by fitting it to the stress values at optimum sampling points
  - nodal stress values are determined from the polynomial approximation
  - stress is interpolated within elements using eq. *)

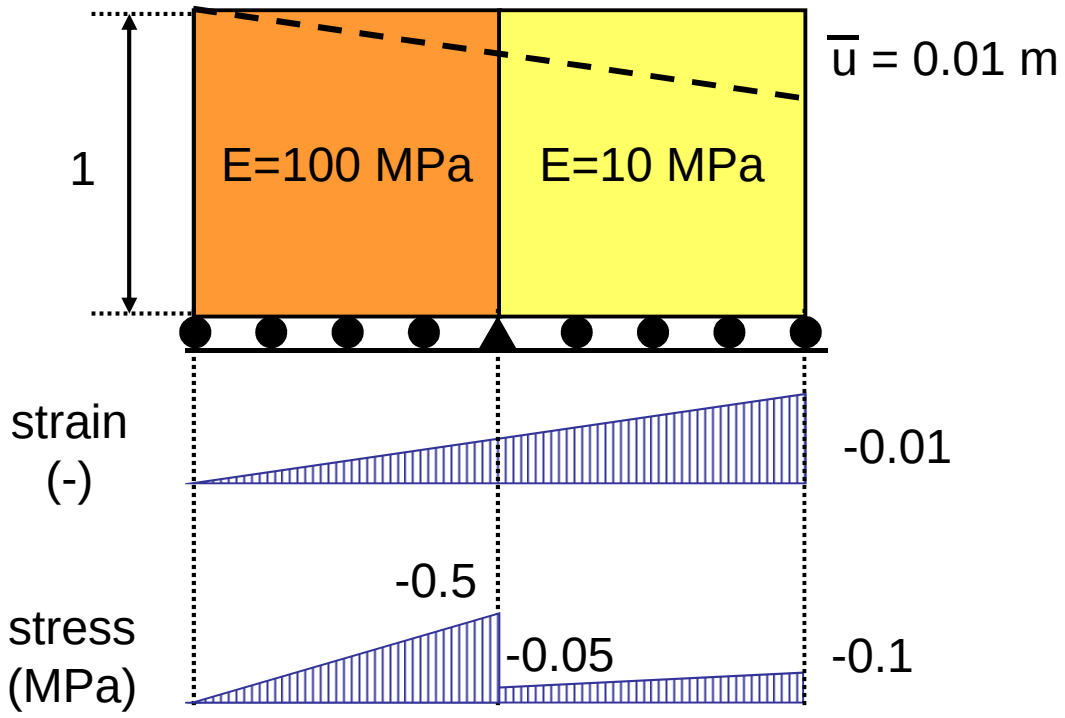


element patch

- nodal values determined from patch
- patch assembly points
- superconvergent sampling points

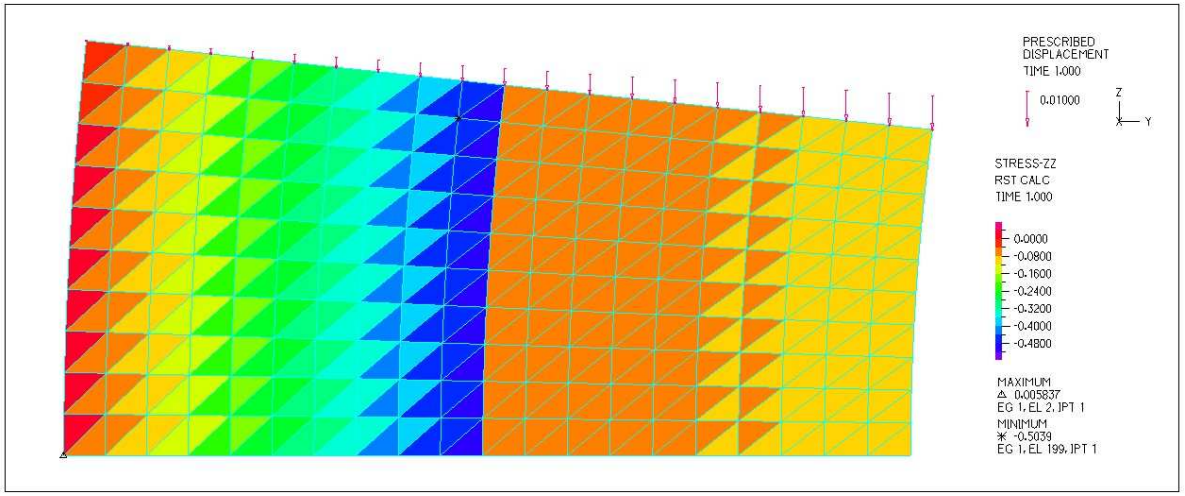
... suitable for regular and irregular meshes, triangular elements.

- The stress/strain recovery/smoothing should be used with caution:
  - in proximity of internal boundaries (interfaces of different materials) where the exact stress or strain may be discontinuous
  - near outer boundaries, where extrapolation from optimum sampling points must be used
  - near locations of stress or strain singularities (concentrated forces, point supports, sharp corners, crack tips,...)
  - example: see next slides

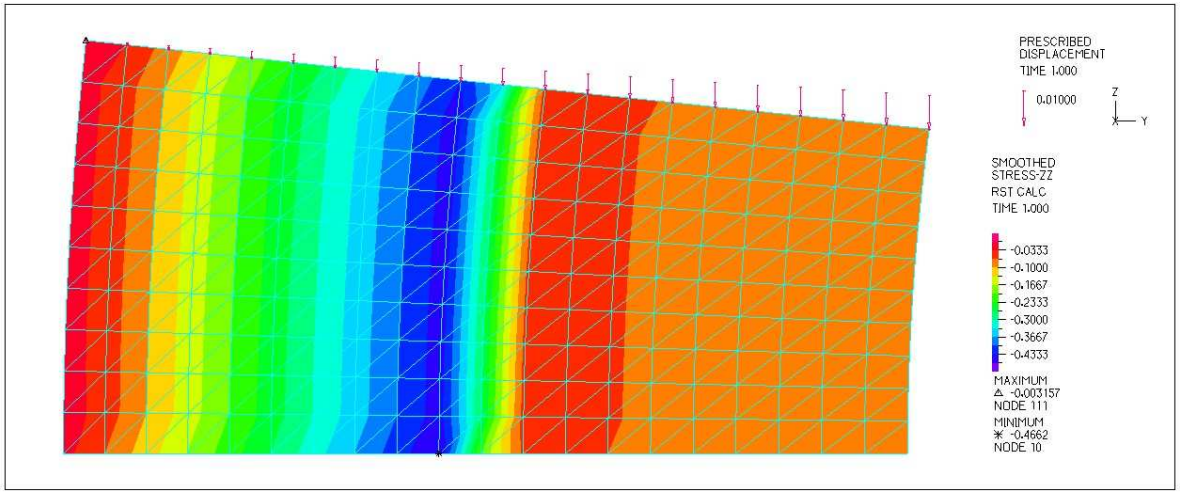

$\bar{u}$ = 0.01 m
1
E=100 MPa
E=10 MPa
strain
(-)
-0.01
-0.5
stress
(MPa)
-0.05
-0.1

Stress plots

No smoothing
- values at element integration point are used



Smoothing applied
- values from integ. points are extrapolated to nodes
- nodal values are averaged
- interpolation *) is used in elements



References

I. Shames & C. Dym: Energy and Finite Element Methods in Structural Mechanics, Taylor & Francis, 1991

K.J. Bathe: Finite Element Procedures, Prentice Hall, Inc., 1996

ADINA R&D, Inc.: Theory and modeling guide, Volume I: ADINA, November 2006

O.C. Zienkiewicz & R.L. Taylor: The Finite Element Method, Vol. 1 – The Basis, Butterworth-Heinemann, 2000

Remark

This document is designated solely as a teaching aid for students of CTU in Prague, Faculty of Civil Engineering, course Numerické metody v inženýrských úlohách.

This document is being continuously updated and corrected by the author. Despite author's utmost effort, it may contain inaccuracies and errors.

Limitation on Liability. Except to the extent required by applicable law, in no event will the author be liable to any user of this document on any legal theory for any special, incidental, consequential, punitive or exemplary damages arising out of the use of the work, even if author has been advised of the possibility of such damages.

Last modified: 28.11.2012