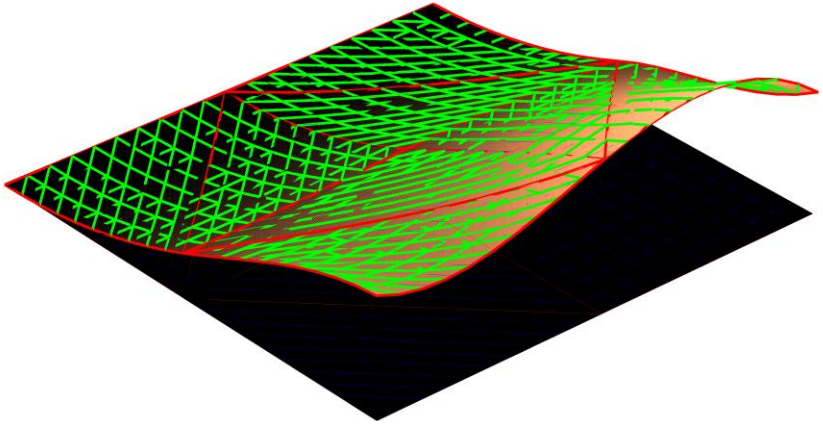
Plates and Shells

Idea of these lectures

- ❑ Make the students familiar with the finite element theory behind standard plates and shells
- ❑ Through exercises make the students able to program various plate and shell elements in Matlab

- ❑ When the lectures are finished, the students should have made a working Matlab program for solving finite element problems using plate and shell elements.



Lecture plan

- ❑ Today
  - ◆ Repetition: steps in the Finite Element Method (FEM)
  - ◆ General steps in a Finite Element program
    - ❑ Investigate the existing Matlab program
  - ◆ Theory of a Kirchhoff plate element
    - ❑ Strong formulation
    - ❑ Weak formulation
  - ◆ Changes in the program when using 3-node Kirchhoff plate elements
  - ◆ Area coordinates
    - ❑ Gauss quadrature using area coordinates
  - ◆ Shape functions for 3-node element
    - ❑ N- and B-matrix for 3-node Kirchhoff plate element
  - ◆ Transformation of degrees of freedom and stiffness matrix
  - ◆ How to include the inplane constant-strain element into the formulation
  - ◆ Laminated plates of orthotropic material

Lecture plan

- Lectures 3+4 (LA)
  - Degenerate 3-D continuum element
  - Thick plates and curved shells
- Lecture 5 (SRKN)
  - Various shell formulations
  - Geometry of curved surfaces

The finite-element method (FEM)

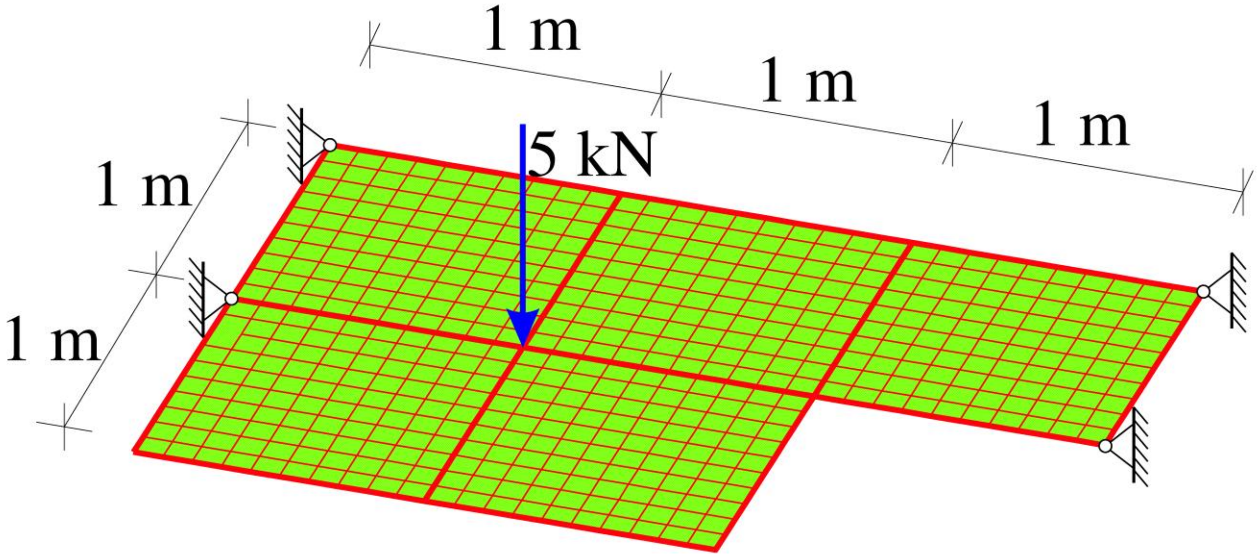
- ❑ Basic steps of the displacement-based FEM
  - ◆ Establish strong formulation
  - ◆ Establish weak formulation
  - ◆ Discretize over space
  - ◆ Select shape and weight functions
  - ◆ Compute element matrices
  - ◆ Assemble global system of equations
  - ◆ Apply nodal forces/forced displacements
  - ◆ Solve global system of equations
  - ◆ Compute stresses/strains etc.

Exercise 1

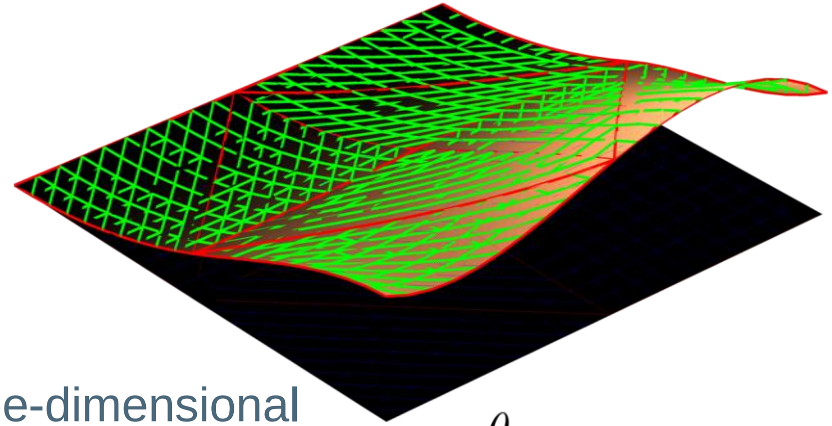
- ☐ How do we make a Finite Element program?
  - ◆ What do we need to define? Pre-processing.
  - ◆ What are the steps in solving the finite element problem? Analysis.
  - ◆ What kind of output are we interested in? Post-processing.

Exercise 2

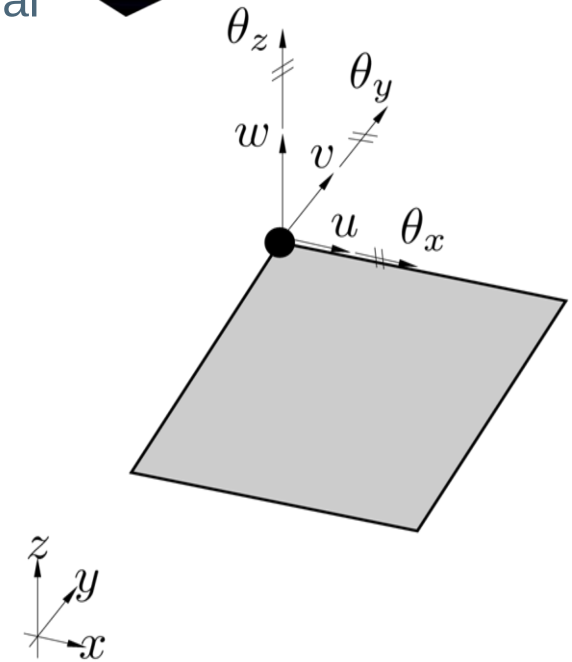
- ☐ Look through the program
- ☐ Determine where the steps discussed in exercise 1 are defined or calculated in the program
- ☐ Try to solve the deformation for the following setup using conforming and non-conforming 4-node elements



What is a plate?

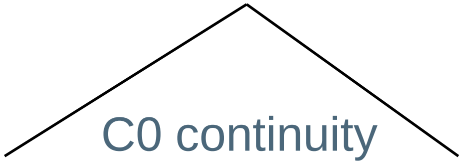


- A plate is a particular form of a three-dimensional solid with a thickness very small compared with other dimensions.
- Today we look at elements with 6 degrees of freedom at each node
  - 3 translations (u,v,w) and 3 rotations (θ_x, θ_y, θ_z)
  - Plate part (w, θ_x, θ_y)
  - in-plane (u,v)
  - zero stiffness (θ_z)
- We distinguish between thin plate theory (Kirchhoff) and thick plate theory (Mindlin-Reissner)



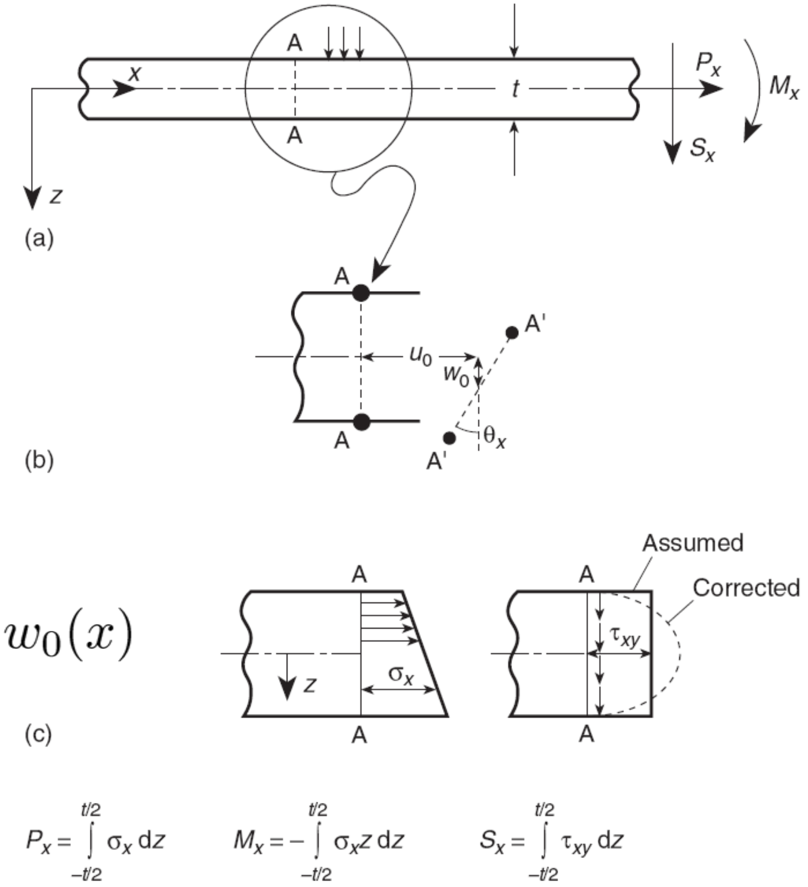
Thin plate theory

- First we assume isotropic homogenous material, i.e. in-plane and out-of-plane components are decoupled
- Only considering the out-of-plane deformations, it is possible to represent the state of deformation by one quantity, w (lateral displacement of the middle plane of the plate)
- This introduces, as we will see later, second derivatives of w in the strain description. (Euler-Bernoulli beam theory)
- Hence, continuity of both the quantity and the derivative across elements are necessary for the second derivative not to vanish (C1 continuity).



Strong formulation of the plate problem (thin and thick plates)

- ☐ Assumptions (first 2D for simplification)
  - ◆ Plane cross sections remain plane
  - ◆ The stresses in the normal direction, z, are small, i.e. strains in that direction can be neglected
- ☐ This implies that the state of deformation is described by



$$u(x,z) = u_0(x) - z\theta_x(x), \quad w(x,z) = w_0(x)$$

Strain and stress components

- Deformations

$$u(x,z) = u_0(x) - z\theta_x(x), \quad w(x,z) = w_0(x)$$

- Strains

$$\varepsilon_x = \frac{\partial u}{\partial x} = \frac{\partial u_0}{\partial x} - z\frac{\partial \theta_x}{\partial x}, \quad \varepsilon_z = 0, \quad \gamma_{xz} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} = -\theta_x + \frac{\partial w_0}{\partial x}$$

- Stresses

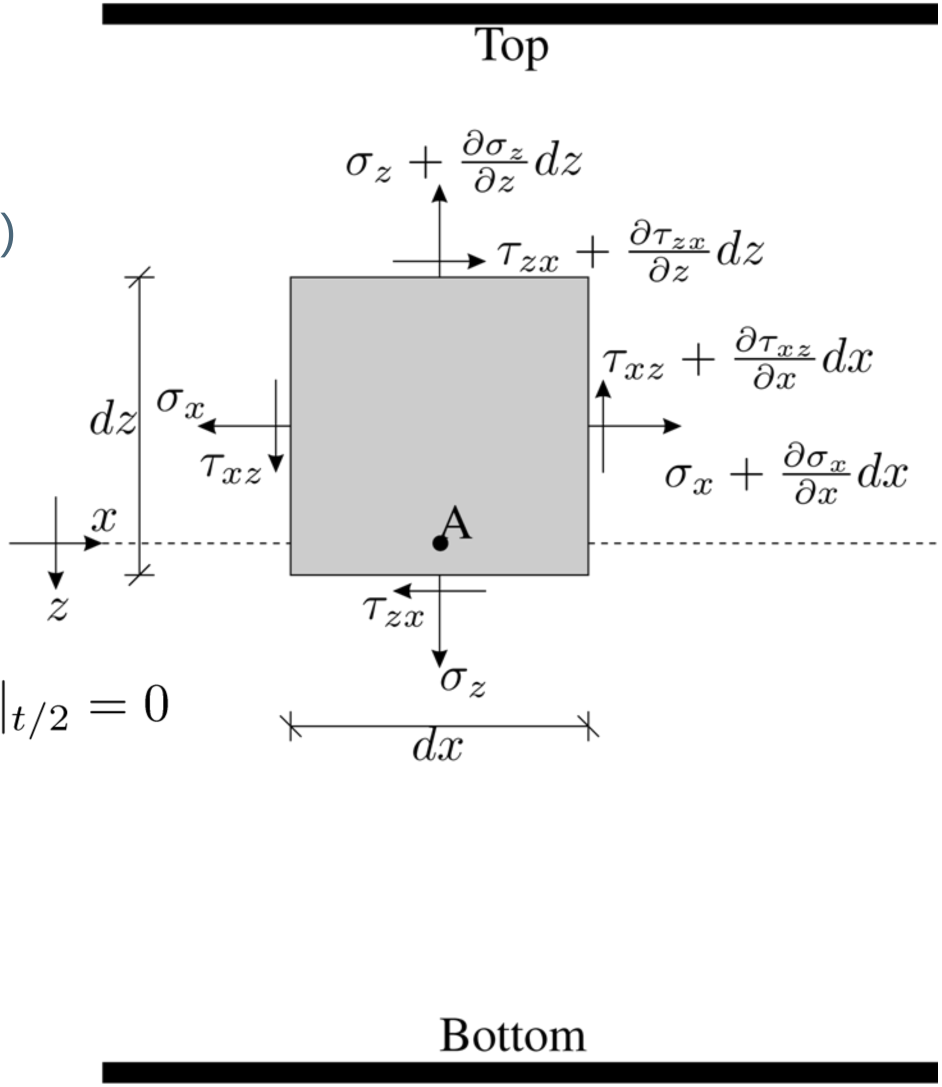
$$\sigma_x = \frac{E}{1-\nu^2}\varepsilon_x, \quad \tau_{xy} = G\gamma_{xy}$$

- Stress resultants (section forces)

$$P_x = \int_{-t/2}^{t/2} \sigma_x dz, \quad S_x = \int_{-t/2}^{t/2} \tau_{xy} dz, \quad M_x = -\int_{-t/2}^{t/2} z\sigma_x dz$$

Equilibrium equations

- Horizontal equilibrium (+right)



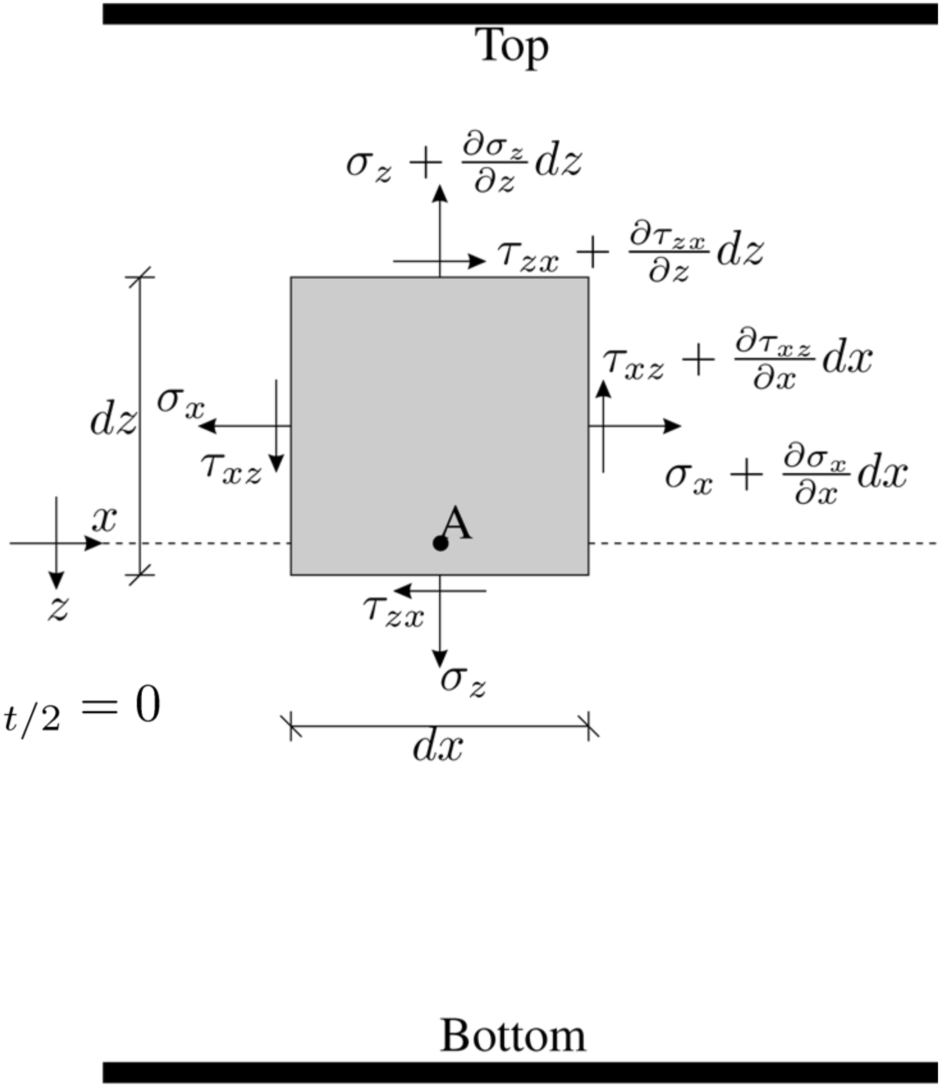
$$\int_{-t/2}^{t/2}\left(\frac{\partial\sigma_x}{\partial x}+\frac{\partial\tau_{zx}}{\partial z}\right)dz=0$$

$$\frac{\partial}{\partial x}\int_{-t/2}^{t/2}\sigma_x dz+\tau_{zx}|_{-t/2}-\tau_{zx}|_{t/2}=0$$

$$\boxed{\frac{\partial P_x}{\partial x}=0}$$

Equilibrium equations

- Vertical equilibrium (+up)



$$\int_{-t/2}^{t/2}\left(\frac{\partial \tau_{xz}}{\partial x}+\frac{\partial \sigma_z}{\partial z}\right)dz = 0$$

$$\frac{\partial}{\partial x}\int_{-t/2}^{t/2}\tau_{xz}dz + \sigma_z|_{-t/2} - \sigma_z|_{t/2} = 0$$

$$\boxed{\frac{\partial S_x}{\partial x}+q_z=0}$$

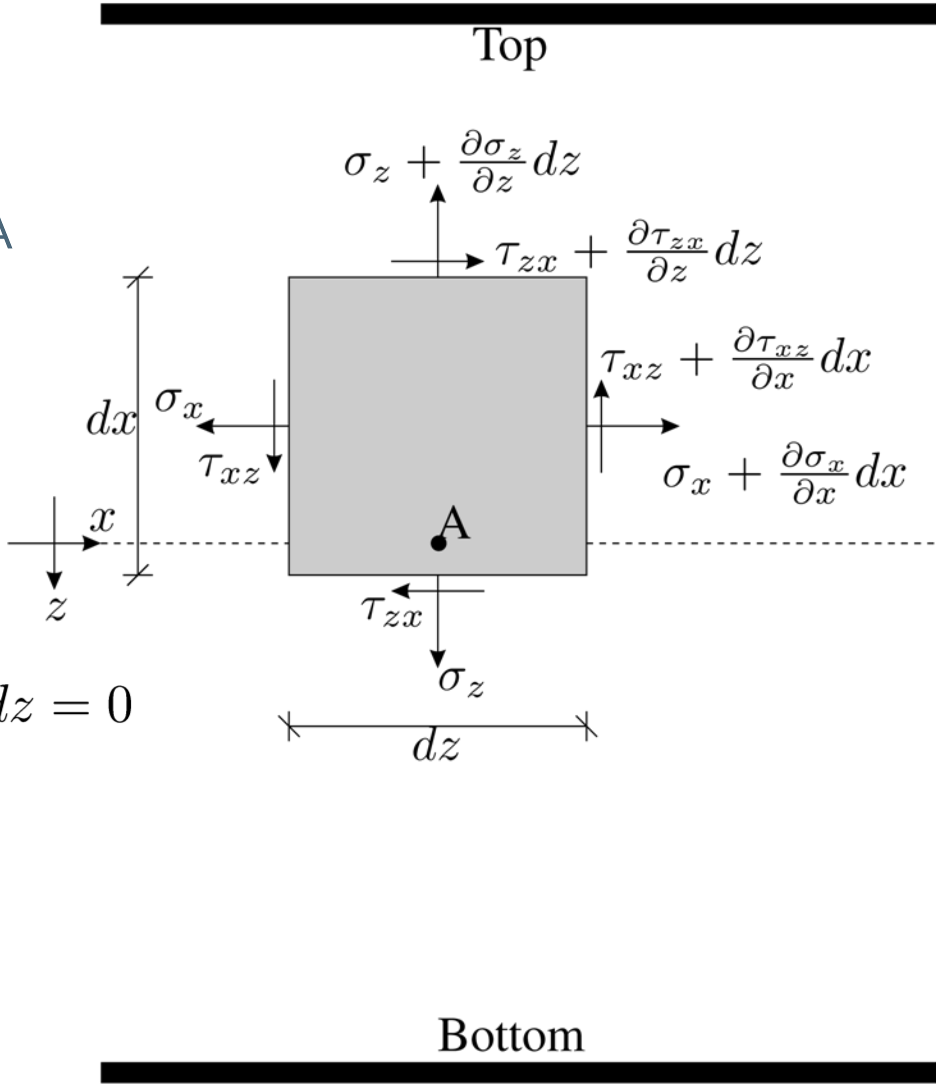
Equilibrium equations

- ☐ Moment equilibrium around A (+clockwise)

$$-\int_{-t/2}^{t/2} z\left(\frac{\partial\sigma_x}{\partial x}+\frac{\partial\tau_{zx}}{\partial z}\right)dz=0$$

$$-\frac{\partial}{\partial x}\int_{-t/2}^{t/2} z\sigma_z dz+\int_{-t/2}^{t/2}\tau_{zx}dz=0$$

$$\boxed{\frac{\partial M_x}{\partial x}+S_x=0}$$



Stress resultants in terms of deformation components

- ☐ Normal force

$$P_x = \int_{-t/2}^{t/2} \sigma_x dz = \frac{E}{1-\nu^2} \int_{-t/2}^{t/2} \varepsilon_x dz = \frac{Et}{1-\nu^2} \frac{\partial u_0}{\partial x}$$

$$\varepsilon_x = \frac{\partial u_0}{\partial x} - z \frac{\partial \theta_x}{\partial x}, \quad \sigma_x = \frac{E}{1-\nu^2} \varepsilon_x$$

- ☐ Shear force

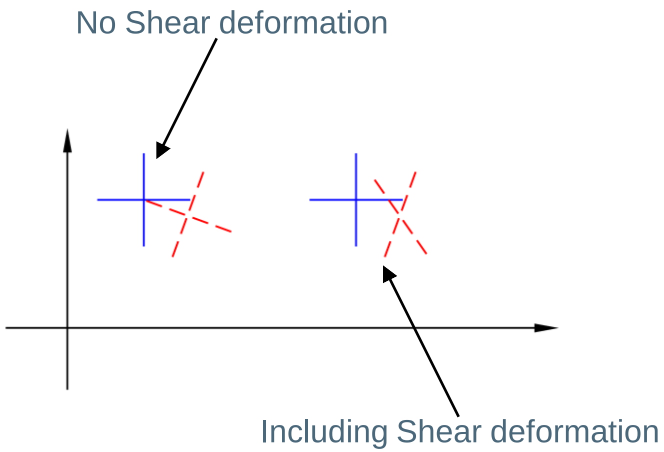
$$S_x = \int_{-t/2}^{t/2} \tau_{xy} dz = \kappa G t \left(\frac{\partial w_0}{\partial x} - \theta_x \right), \quad \kappa = 5/6$$

Rectangular cross section

- ☐ Moment

$$M_x = \int_{-t/2}^{t/2} z \sigma_x dz = \frac{Et^3}{12(1-\nu^2)} \frac{\partial \theta_x}{\partial x}$$

Thin plate approximation



- ☐ Neglects the shear deformation, G= ∞

$$S_x = \kappa G t \left(\frac{\partial w_0}{\partial x} - \theta_x \right)$$

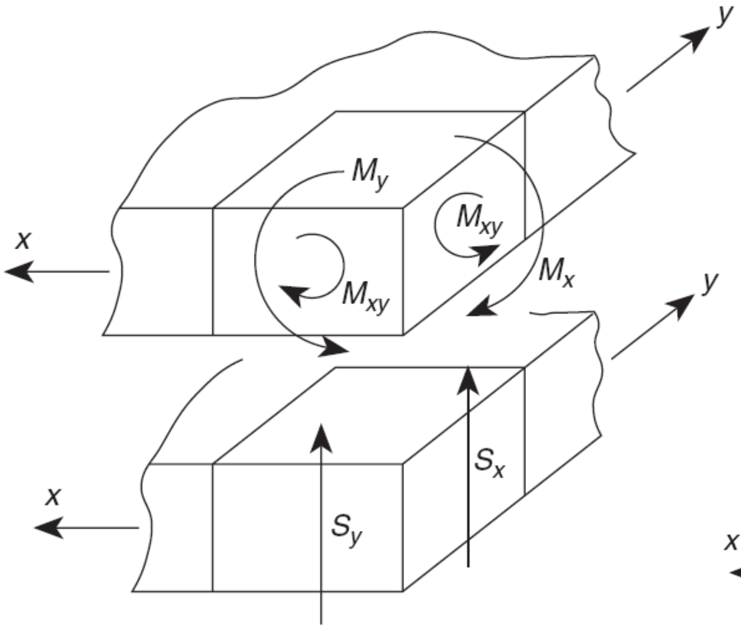
- ☐ The shear force should not introduce infinite energy into the system, hence

$$\boxed{\frac{\partial w_0}{\partial x} - \theta_x = 0}$$

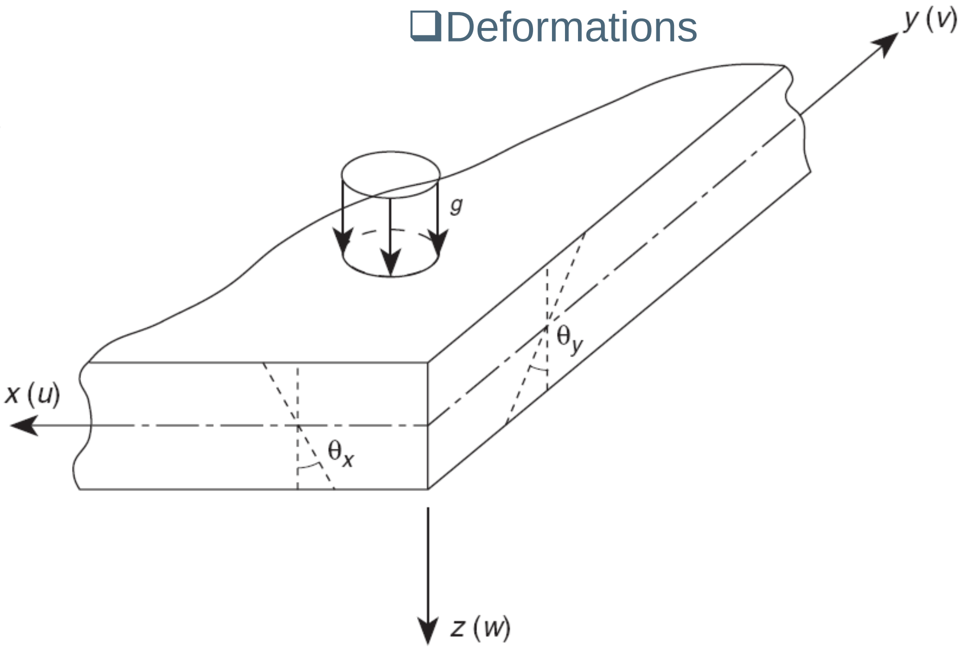
- ☐ I.e. rotations can be determined from the bending displacement

General three-dimensional case (disregarding inplane deformations)

- ❑Forces



- ❑Deformations



Kinematic relations

- ❑ Deformations

NB! $\theta_x = -\dfrac{\partial u}{\partial z}, \theta_y = -\dfrac{\partial v}{\partial z}$ (not "physical" rotations)

$$u = -z\theta_x(x,y), \quad v = -z\theta_y(x,y), \quad w = w_0(x,y)$$

- ❑ Strains

See figure slide 16

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = -z \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} \theta_x \\ \theta_y \end{bmatrix} = -z\mathbf{L}\boldsymbol{\theta}, \quad \mathbf{L} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}$$

$$\boldsymbol{\gamma} = \begin{bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{bmatrix} = \begin{bmatrix} \frac{\partial w}{\partial x} \\ \frac{\partial w}{\partial y} \end{bmatrix} - \begin{bmatrix} \theta_x \\ \theta_y \end{bmatrix} = \boldsymbol{\nabla} w - \boldsymbol{\theta}, \quad \boldsymbol{\nabla} = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix}$$

See slide 15

Constitutive relation

☐ Isotropic, linear elastic material

$$\begin{bmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{bmatrix} = \frac{E}{1-\nu^2}\begin{bmatrix}1 & \nu & 0\\ \nu & 1 & 0\\ 0 & 0 & (1-\nu)/2\end{bmatrix}\begin{bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{bmatrix} = \mathbf{E}\boldsymbol{\varepsilon}$$

Section moments and shear forces

- ☐ Moments

$$M_x = -\int_{-t/2}^{t/2} z\sigma_x dz, \quad M_y = -\int_{-t/2}^{t/2} z\sigma_y dz, \quad M_{xy} = -\int_{-t/2}^{t/2} z\tau_{xy} dz$$

- ☐ Using the constitutive (slide 18) and kinematic (slide 17) relations we get

$$\mathbf{M} = \begin{bmatrix} M_x \\ M_y \\ M_{xy} \end{bmatrix} = \mathbf{DL}\boldsymbol{\theta}, \quad \mathbf{D} = \int_{-t/2}^{t/2} z^2 \mathbf{E} dz$$

- ☐ Shear forces

$$\mathbf{S} = \begin{bmatrix} S_x \\ S_y \end{bmatrix} = \boldsymbol{\alpha}(\boldsymbol{\nabla} w - \boldsymbol{\theta}), \quad \boldsymbol{\alpha} = \kappa G t \mathbf{I}$$

Equilibrium equations

- ☐ 2D

$$\boxed{\frac{\partial M_x}{\partial x} + S_x = 0} \qquad \boxed{\frac{\partial S_x}{\partial x} + q_z = 0}$$

- ☐ 3D

$$\boxed{\mathbf{L}^T\mathbf{M} + \mathbf{S} = \mathbf{0}} \qquad \boxed{\boldsymbol{\nabla}^T\mathbf{S} + q = 0}$$

- ☐ Combining

$$\boldsymbol{\nabla}^T\mathbf{L}^T\mathbf{M} - q = 0$$

Plates and Shells

Thin plates

- ☐ Shear deformations out of plane are disregarded, I.e.

$$\gamma = \mathbf{0} = \boldsymbol{\nabla}\mathbf{w} - \boldsymbol{\theta}$$

$$\boldsymbol{\varepsilon} = -z\mathbf{L}\boldsymbol{\theta} = -z\mathbf{L}\boldsymbol{\nabla}\mathbf{w} = -z\begin{bmatrix} \frac{\partial^2 w}{\partial x^2} \\ \frac{\partial^2 w}{\partial y^2} \\ 2\frac{\partial^2 w}{\partial x \partial y} \end{bmatrix}$$

- ☐ Equilibrium equation (strong formulation of the thin plate)

$$\boldsymbol{\nabla}^T\mathbf{L}^T\mathbf{M} - q = 0, \qquad \mathbf{M} = \mathbf{D}\mathbf{L}\boldsymbol{\theta} = \mathbf{D}\mathbf{L}\boldsymbol{\nabla}w$$

$$\boldsymbol{\nabla}^T\mathbf{L}^T\mathbf{D}\mathbf{L}\boldsymbol{\nabla}w - q = 0,$$

Weak formulation (Principle of virtual work)

- ☐ Internal virtual work

Definition

$$\delta\Pi_{int} = \int_{\Omega} (\delta\boldsymbol{\varepsilon})^T \boldsymbol{\sigma}\, d\Omega = \int_{\Omega} \delta w (\mathbf{L}\boldsymbol{\nabla})^T \mathbf{D} (\mathbf{L}\boldsymbol{\nabla}) w\, d\Omega$$

- ☐ External virtual work

$$\delta\Pi_{ext} = \int_{\Omega} \delta w q\, d\Omega + \sum_i \delta w_i R_i + \int_{\Gamma_n} \delta\theta_s \left(\bar{S}_n - \frac{\bar{M}_{ns}}{\partial s} \right) , d\Gamma$$

distributed load

nodal load

line boundary load

Finite-element formulation

- ☐ Galerkin approach, physical and variational fields are discretised using the same interpolation functions

$$w = \mathbf{N}\mathbf{u}, \quad \delta w = \mathbf{N}\delta\mathbf{u}$$

- ☐ The variation of the sum of internal and external work should be zero for any choice of δ***u***
- ☐ FEM equations

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \quad \mathbf{u} = [u_1 \, v_1 \, w_1 \, \theta_{x1} \, \theta_{y1} \, \theta_{z1} \, u_2 \, \ldots]^T$$

$$\mathbf{K} = \int_\Omega \mathbf{B}^T\mathbf{D}\mathbf{B}\, d\Omega \quad \mathbf{B} = (\mathbf{L}\boldsymbol{\nabla})\mathbf{N}$$

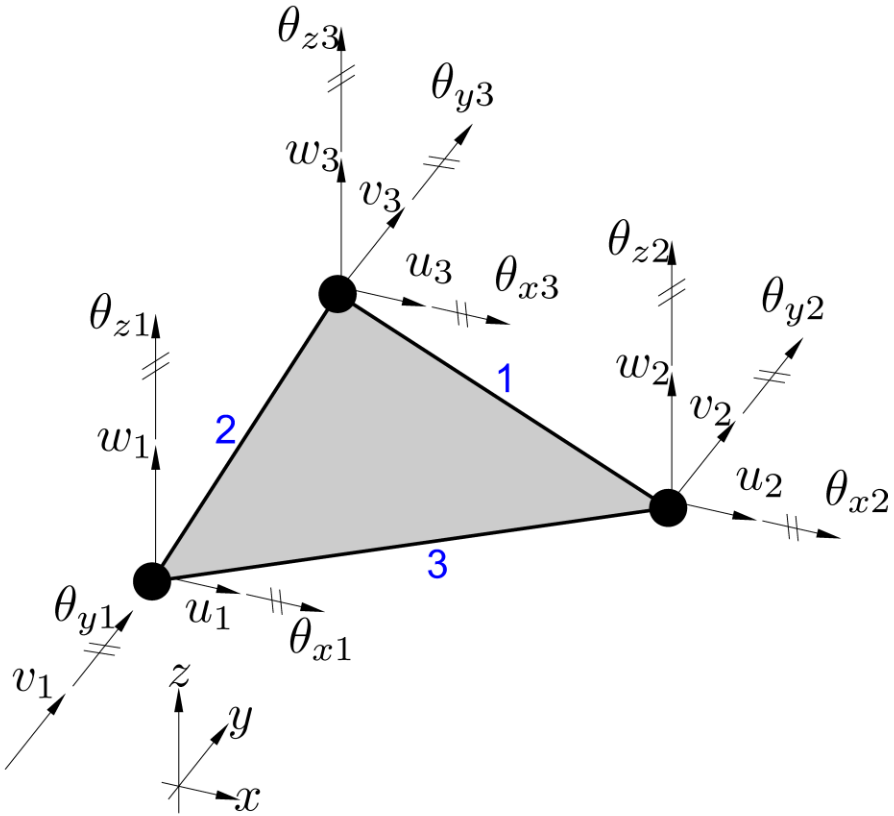
$$\mathbf{f} = \int_\Omega \mathbf{N}^T q \, d\Omega + \mathbf{R}$$

nodal load

Consistent area load

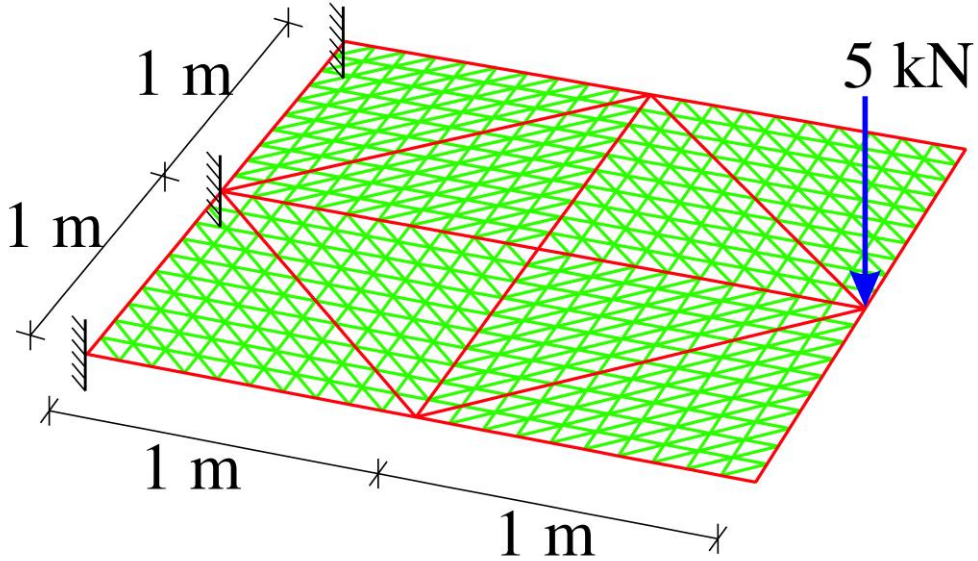
Triangular elements

- ☐ 3 Nodes, 6 global degrees of freedom per node



Exercise 3

- ☐ What do we need to change in the program when using 3-node elements (6 global DOF per node) compared with 4-node elements (6 global DOF per node)?
- ☐ Make the following setup using 3-node elements



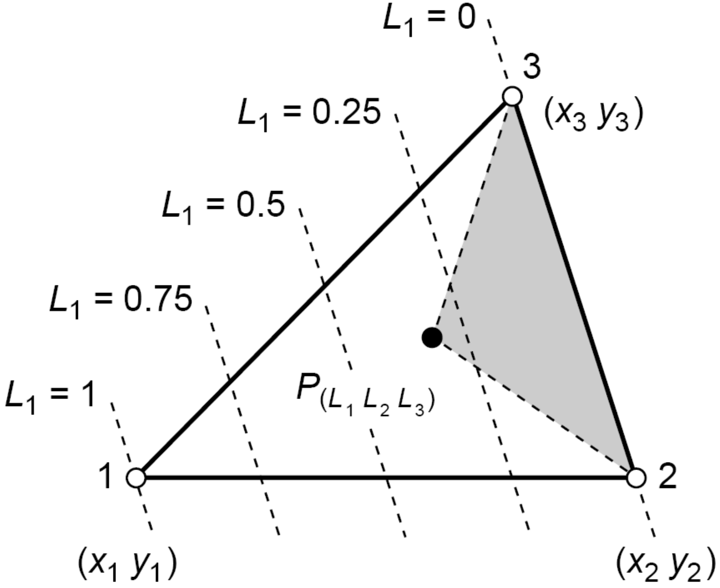
Triangular elements, Area coordinates

- A set of coordinates L_1, L_2 and L_3 are introduced, given as

$$x = L_1 x_1 + L_2 x_2 + L_3 x_3$$

$$y = L_1 y_1 + L_2 y_2 + L_3 y_3$$

$$1 = L_1 + L_2 + L_3$$



- Alternatively

$$L_1 = \frac{\text{area}P23}{\text{area}123}, \quad L_2 = \frac{\text{area}P31}{\text{area}123}, \quad L_1 = \frac{\text{area}P21}{\text{area}123}$$

Triangular elements, Area coordinates

- ☐ Area coordinates in terms of Cartesian coordinates

$$L_1 = \frac{a_1 + b_1 x + c_1 y}{2\Delta}, \qquad L_2 = \frac{a_2 + b_2 x + c_2 y}{2\Delta}, \qquad L_3 = \frac{a_3 + b_3 x + c_3 y}{2\Delta}$$

$$a_1 = x_2 y_3 - x_3 y_2 \qquad a_2 = x_3 y_1 - x_1 y_3 \qquad a_3 = x_1 y_2 - x_2 y_1$$

$$b_1 = y_2 - y_3 \qquad b_2 = y_3 - y_1 \qquad b_3 = y_1 - y_2$$

$$c_1 = x_3 - x_2 \qquad c_2 = x_1 - x_3 \qquad c_3 = x_2 - x_1$$

$$\Delta = \text{area } 123 = \frac{1}{2}(b_1 c_2 - b_2 c_1)$$

- ☐ In compact form

$$L_i = \frac{a_i + b_i x + c_i y}{2\Delta}, \qquad \begin{aligned} a_i &= x_j y_k - x_k y_j \\ b_i &= y_j - y_k \\ c_i &= x_k - x_j \end{aligned}$$

$i = 1, 2, 3$

i, j, k as positive cyclic permutation

Shape functions (only out-of-plane components considered)

- ☐ First index indicates the node, second index indicates the DOF

$$[w] = \begin{bmatrix} N_{11} & N_{12} & N_{13} & N_{21} & N_{22} & N_{23} & N_{31} & N_{32} & N_{33} \end{bmatrix} \begin{bmatrix} w_1 \\ \theta_{x1} \\ \theta_{y1} \\ w_2 \\ \theta_{x2} \\ \theta_{y2} \\ w_3 \\ \theta_{x3} \\ \theta_{y3} \end{bmatrix}$$

$$\mathbf{N}^T = \begin{bmatrix} P_1 - P_4 + P_6 + 2(P_7 - P_9) \\ -b_2(P_9 - P_6) - b_3 P_7 \\ -c_2(P_9 - P_6) - c_3 P_7 \\ P_2 - P_5 + P_4 + 2(P_8 - P_7) \\ -b_3(P_7 - P_4) - b_1 P_8 \\ -c_3(P_7 - P_4) - c_1 P_8 \\ P_3 - P_6 + P_5 + 2(P_9 - P_8) \\ -b_1(P_8 - P_5) - b_2 P_9 \\ -c_1(P_8 - P_5) - c_2 P_9 \end{bmatrix} \qquad \mu_i = \frac{l_k^2 - l_j^2}{l_i^2}, \qquad l_i = \text{length of side } i$$

$$\mathbf{P} = [L_1, L_2, L_3, L_1 L_2, L_2 L_3, L_3 L_1,$$

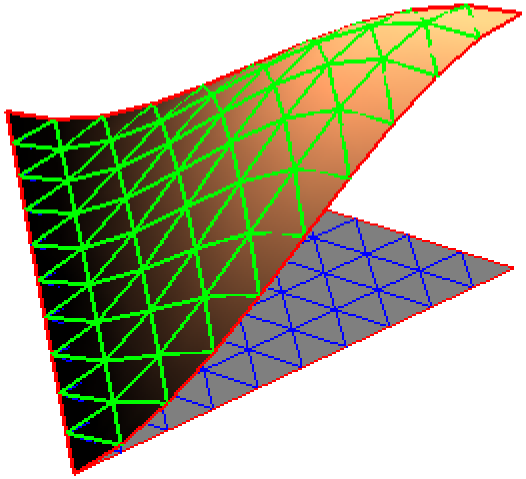
$$L_1^2 L_2 + \frac{1}{2} L_1 L_2 L_3 \big(3(1 - \mu_3) L_1 - (1 + 3\mu_3) L_2 + (1 + 3\mu_3) L_3\big)$$

$$L_2^2 L_3 + \frac{1}{2} L_1 L_2 L_3 \big(3(1 - \mu_1) L_2 - (1 + 3\mu_1) L_3 + (1 + 3\mu_1) L_1\big)$$

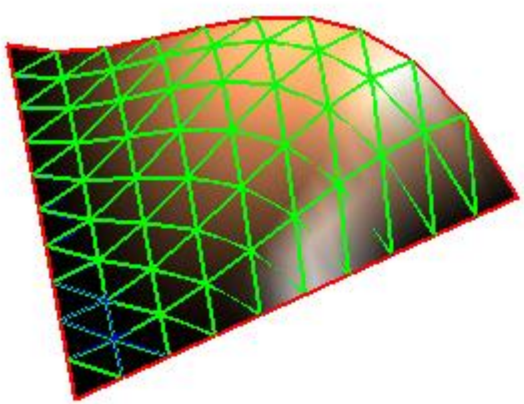
$$L_3^2 L_1 + \frac{1}{2} L_1 L_2 L_3 \big(3(1 - \mu_2) L_3 - (1 + 3\mu_2) L_1 + (1 + 3\mu_2) L_2\big)]$$

Shape functions

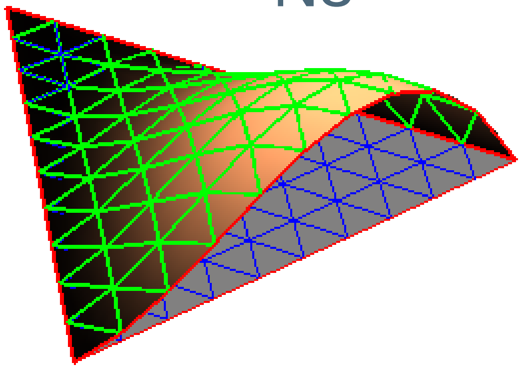
N1



N2



N3



Exercise 4

- Program the shape functions

```
function [N, B] = shape_trinagular3n(L,xe) ;
```

  - input: 3 area coordinates, 3 local node coordinates

```
L = [L1 L2 L3];

xe = [x1 y1 z1 ;
      x2 y2 z2 ;
      x3 y3 z3 ];
```

  - output: shapefunctions organised in the following way (size(N) = [3x15])

```
% Shape function
N = [0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   ;
     0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   ;
     0   0   N11 N12 N13 0   0   N21 N22 N23 0   0   N31 N32 N33 ];
```

B-matrix

- ☐ For the out-of-plane part, **B** is the second derivative (with respect to *x* and *y*) of the shape functions

$$\mathbf{K} = \int_\Omega \mathbf{B}^T \mathbf{D} \mathbf{B} \, d\Omega \qquad \mathbf{B} = (\mathbf{L}\boldsymbol{\nabla})\mathbf{N}$$

$$\mathbf{B} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \mathbf{N} = \begin{bmatrix} \frac{\partial^2}{\partial x^2} \\ \frac{\partial^2}{\partial y^2} \\ 2\frac{\partial^2}{\partial x \partial y} \end{bmatrix} \mathbf{N}$$

Derivative with respect to L_1, L_2 and L_3

$$L_i = \frac{a_i + b_i x + c_i y}{2\Delta},$$

- First order derivatives

$$\begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial L_1}{\partial x} & \frac{\partial L_2}{\partial x} & \frac{\partial L_3}{\partial x} \\ \frac{\partial L_1}{\partial y} & \frac{\partial L_2}{\partial y} & \frac{\partial L_3}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial L_1} \\ \frac{\partial}{\partial L_2} \\ \frac{\partial}{\partial L_3} \end{bmatrix} = \frac{1}{2\Delta} \begin{bmatrix} b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial L_1} \\ \frac{\partial}{\partial L_2} \\ \frac{\partial}{\partial L_3} \end{bmatrix}$$

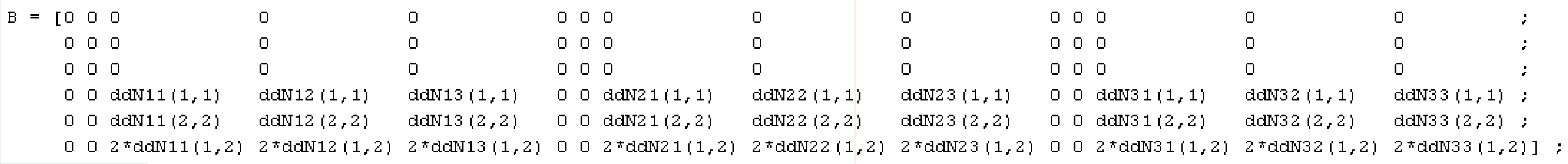
- Second order derivatives

$$\begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial^2}{\partial x^2} & \frac{\partial^2}{\partial x \partial y} \\ \frac{\partial^2}{\partial x \partial y} & \frac{\partial^2}{\partial y^2} \end{bmatrix} =$$

$$\boxed{\frac{1}{4\Delta^2} \begin{bmatrix} b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \begin{bmatrix} \frac{\partial^2}{\partial L_1^2} & \frac{\partial^2}{\partial L_1 \partial L_2} & \frac{\partial^2}{\partial L_1 \partial L_3} \\ \frac{\partial^2}{\partial L_2 \partial L_1} & \frac{\partial^2}{\partial L_2^2} & \frac{\partial^2}{\partial L_2 \partial L_3} \\ \frac{\partial^2}{\partial L_3 \partial L_1} & \frac{\partial^2}{\partial L_3 \partial L_2} & \frac{\partial^2}{\partial L_3^2} \end{bmatrix} \begin{bmatrix} b_1 & c_1 \\ b_2 & c_2 \\ b_2 & c_2 \end{bmatrix}}$$

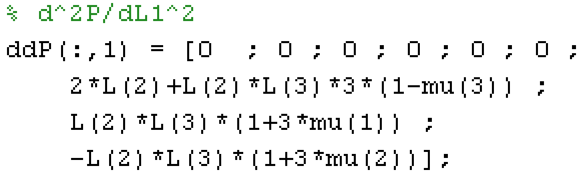
Exercise 5

- ☐ Identify where the B-matrix is created
  - ◆ The B-matrix should be organized as follows



```
B = [0  0  0             0             0             0  0  0             0             0             0  0  0             0             0             ;
     0  0  0             0             0             0  0  0             0             0             0  0  0             0             0             ;
     0  0  0             0             0             0  0  0             0             0             0  0  0             0             0             ;
     0  0  ddN11(1,1)    ddN12(1,1)    ddN13(1,1)    0  0  ddN21(1,1)    ddN22(1,1)    ddN23(1,1)    0  0  ddN31(1,1)    ddN32(1,1)    ddN33(1,1)    ;
     0  0  ddN11(2,2)    ddN12(2,2)    ddN13(2,2)    0  0  ddN21(2,2)    ddN22(2,2)    ddN23(2,2)    0  0  ddN31(2,2)    ddN32(2,2)    ddN33(2,2)    ;
     0  0  2*ddN11(1,2)  2*ddN12(1,2)  2*ddN13(1,2)  0  0  2*ddN21(1,2)  2*ddN22(1,2)  2*ddN23(1,2)  0  0  2*ddN31(1,2)  2*ddN32(1,2)  2*ddN33(1,2)] ;
```

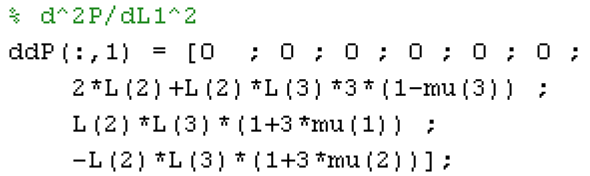
  - ◆ ddNij is the second derivative with respect to *x* and *y* of N_{ij}, where index *i* is the node and *j* is the DOF
  - ◆ make a matrix (9x6) with a row for each shape function and a column for each second order derivative with respect to L_i (e.g. d^2/dL_1^2, d^2/dL_1dL_2,…)



```
% d^2P/dL1^2
ddP(:,1) = [0  ; 0 ; 0 ; 0 ; 0 ; 0 ;
     2*L(2)+L(2)*L(3)*3*(1-mu(3)) ;
     L(2)*L(3)*(1+3*mu(1)) ;
     -L(2)*L(3)*(1+3*mu(2))];
```

Exercise 5 continued

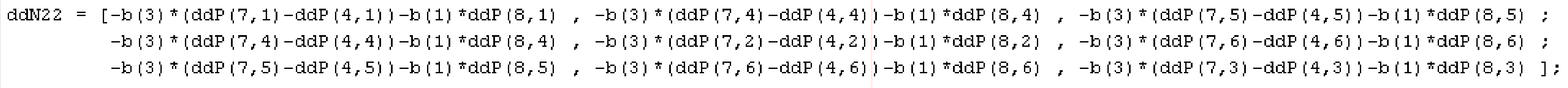
- ☐ make a matrix (9x6) with a row for each shape function and a column for each second order derivative with respect to L_i (e.g. d^2/dL_1^2, d^2/dL_1dL_2,…)



```
% d^2P/dL1^2
ddP(:,1) = [0  ; 0 ; 0 ; 0 ; 0 ; 0 ;
     2*L(2)+L(2)*L(3)*3*(1-mu(3)) ;
     L(2)*L(3)*(1+3*mu(1)) ;
     -L(2)*L(3)*(1+3*mu(2))];
```

N_5 slide 29

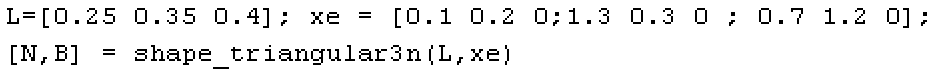
- ☐ Now use the same functions (slide 29) as for the shape functions copied into a 3x3 matrix



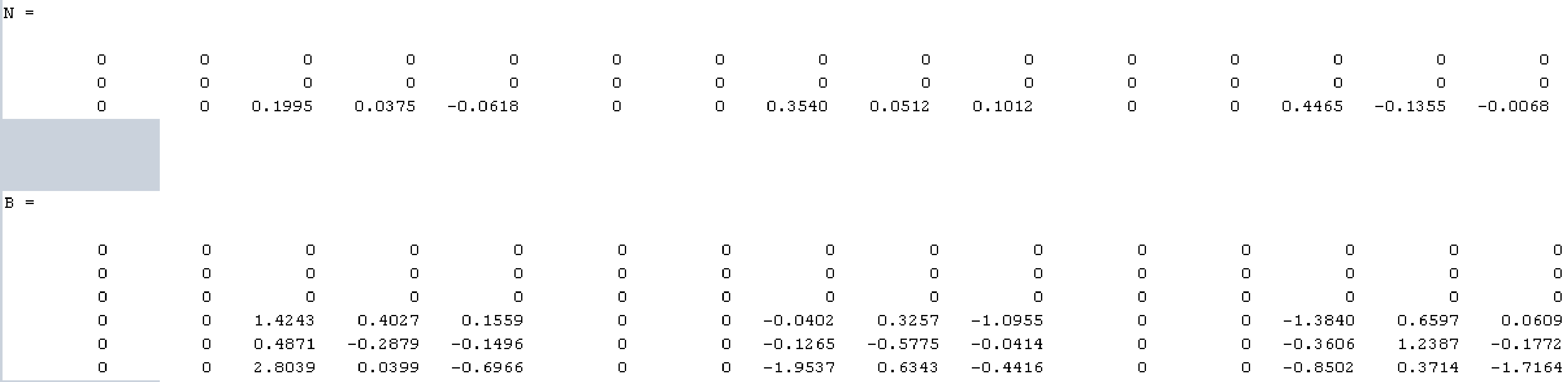
```
ddN22 = [-b(3)*(ddP(7,1)-ddP(4,1))-b(1)*ddP(8,1)  , -b(3)*(ddP(7,4)-ddP(4,4))-b(1)*ddP(8,4)  , -b(3)*(ddP(7,5)-ddP(4,5))-b(1)*ddP(8,5)  ;
         -b(3)*(ddP(7,4)-ddP(4,4))-b(1)*ddP(8,4)  , -b(3)*(ddP(7,2)-ddP(4,2))-b(1)*ddP(8,2)  , -b(3)*(ddP(7,6)-ddP(4,6))-b(1)*ddP(8,6)  ;
         -b(3)*(ddP(7,5)-ddP(4,5))-b(1)*ddP(8,5)  , -b(3)*(ddP(7,6)-ddP(4,6))-b(1)*ddP(8,6)  , -b(3)*(ddP(7,3)-ddP(4,3))-b(1)*ddP(8,3)  ];
```

- ☐ Multiply this with the coefficient matrix to obtain the derivative with respect to *x* and *y*
- ☐ Organize **B** as on the previous slide

Test the shape functions



```
L=[0.25 0.35 0.4]; xe = [0.1 0.2 0;1.3 0.3 0 ; 0.7 1.2 0];
[N,B] = shape_triangular3n(L,xe)
```



```
N =

         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0
         0         0    0.1995    0.0375   -0.0618         0         0    0.3540    0.0512    0.1012         0         0    0.4465   -0.1355   -0.0068


B =

         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0
         0         0    1.4243    0.4027    0.1559         0         0   -0.0402    0.3257   -1.0955         0         0   -1.3840    0.6597    0.0609
         0         0    0.4871   -0.2879   -0.1496         0         0   -0.1265   -0.5775   -0.0414         0         0   -0.3606    1.2387   -0.1772
         0         0    2.8039    0.0399   -0.6966         0         0   -1.9537    0.6343   -0.4416         0         0   -0.8502    0.3714   -1.7164
```

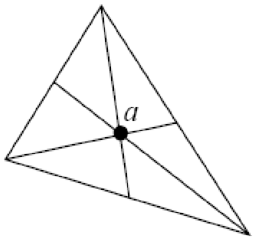
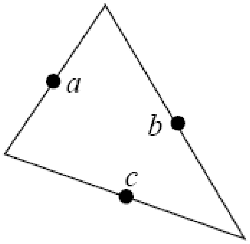
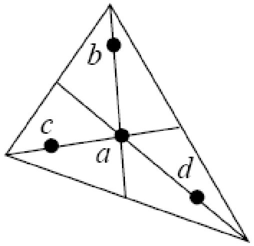
Gauss-quadrature

- Quadrature for solving stiffness integral

$$\mathbf{K} = \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} \, d\Omega \simeq \sum_{i=1}^{n} w_i \mathbf{B}^T (L_{i1}, L_{i2}, L_{i3}) \mathbf{D} \mathbf{B} (L_{i1}, L_{i2}, L_{i3})$$

  - *i* counts over the Gauss-points, w_i are the Gauss weights

Gauss points and weights

Order	Figure	Error	Points	Triangular coordinates	Weights
Linear		$R = O(h^2)$	a	$\frac{1}{3}, \frac{1}{3}, \frac{1}{3}$	1
Quadratic		$R = O(h^3)$	a	$\frac{1}{2}, \frac{1}{2}, 0$	$\frac{1}{3}$
			b	$0, \frac{1}{2}, \frac{1}{2}$	$\frac{1}{3}$
			c	$\frac{1}{2}, 0, \frac{1}{2}$	$\frac{1}{3}$
Cubic		$R = O(h^4)$	a	$\frac{1}{3}, \frac{1}{3}, \frac{1}{3}$	$-\frac{27}{48}$
			b	$0.6, 0.2, 0.2$	$\frac{25}{48}$
			c	$0.2, 0.6, 0.2$	
			d	$0.2, 0.2, 0.6$	

Exercise 6

- Modify the element stiffness function for determining the element stiffness matrix

```
function Ke = Ke_triangular3n(xe,elemMatDat)
```

  - input: element coordinates, material data
  - output: element stiffness matrix (and mass matrix=0)
  - Tip: copy KeMe_plate4c.m and modify
- Use the Gauss-function

```
function [gp,gw] = Gauss3n(n)
```

  - input: number of Gauss-points
  - output: Gauss-point coordinates and weights
- determine the element stiffness matrix of the 3-node element in the following way
  1. Loop over the number of Gauss points
  2. Determine **B** for Gauss point *i*
  3. Calculate the contribution to the integral for Gauss point *i*
  4. Add the contribution to the stiffness matrix
  5. Repeat 2-4 for all Gauss points

Add the drilling DOF

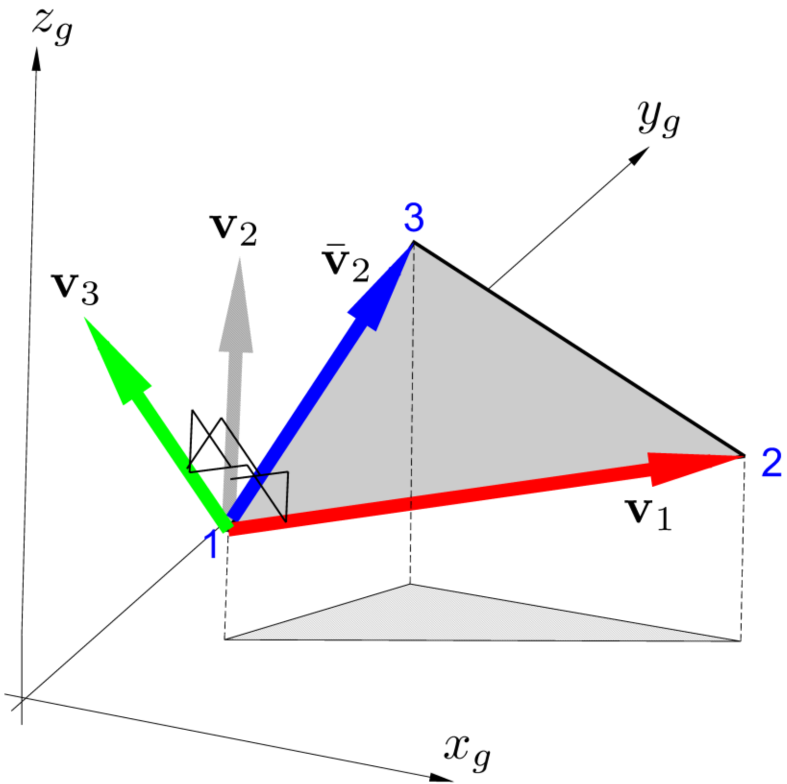
- B-matrix was organized with 6x15, i.e 15 DOF
- In the global system we have 6 DOF per node, i.e. 18 DOF
- We have not included the rotation around the z-axis (the drilling DOF)
- We will not introduce a stiffness for this DOF. But in the global system we need 6 DOF per node
- So we simply introduce a zero stiffness in the element stiffness matrix on row/column 6, 12 and 18
- When the contribution from the Gauss points are added, it is done to the remaining rows,columns, i.e.

```
Kel([1:5 7:11 13:17],[1:5 7:11 13:17]) = Kel([1:5 7:11 13:17],[1:5 7:11 13:17]) + DKe*gw(j) ;
```

- For the problem not to become singular we could introduce an arbitrary stiffness, however this is not necessary to solve the system

Transformation between local and global coordinates

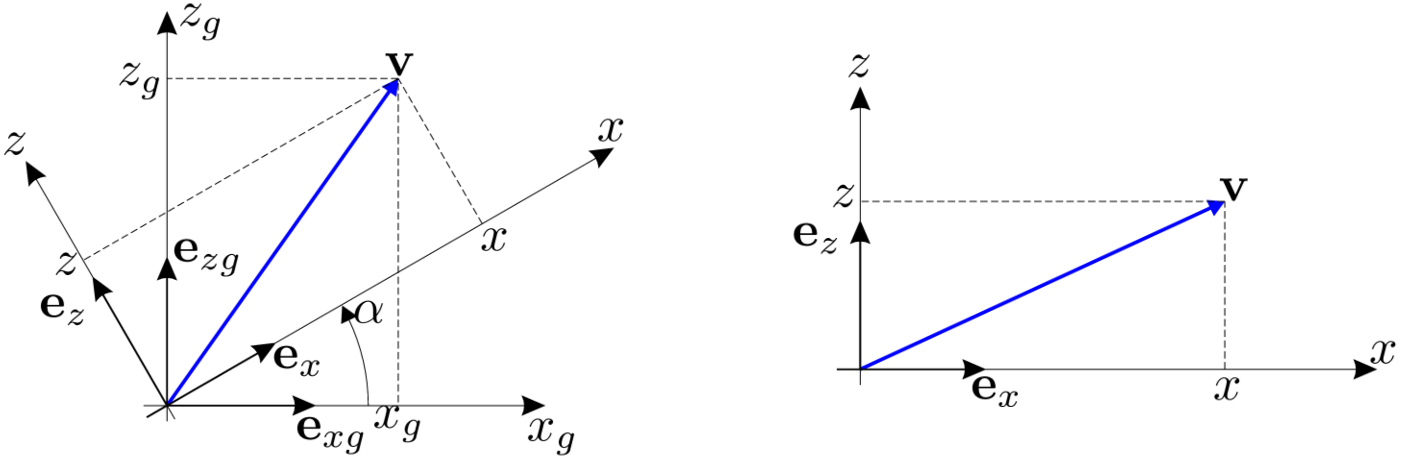
- ☐ The shape functions are defined in the plane of the triangle, i.e. z-coordinates for the nodes are equal to zero



- ☐ We have already identified the local element stiffness matrix $\mathbf{K}_e$, all we need is to determine the transformation matrix **T**

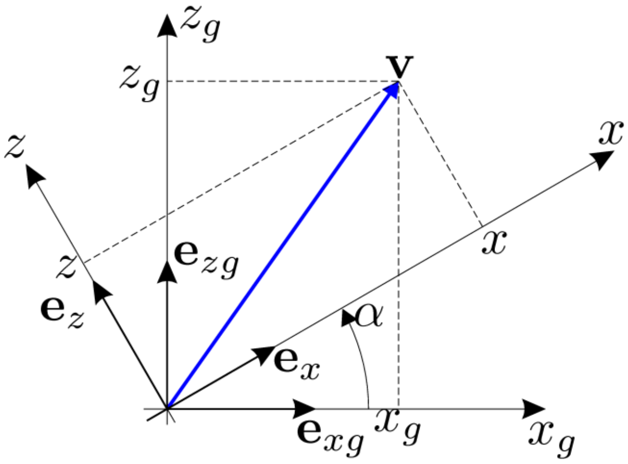
$$\mathbf{K}_{eg} = \mathbf{T}\mathbf{K}_e\mathbf{T}^T$$

- ☐ If we want to describe the components of a vector given in one coordinate system (x_g,y_g) in another coordinate system (x,y), we can multiply the vector with the unit vectors spanning the (x,y) system
- ☐ This corresponds to rotating the vector -α equal the angle between the two systems



Plates and Shells

- ❑ Vectors defined in global system



$$\mathbf{v}_g = \begin{bmatrix} x_g \\ z_g \end{bmatrix}, \quad \mathbf{e}_x = \begin{bmatrix} e_{x1} \\ e_{x2} \end{bmatrix}, \quad \mathbf{e}_z = \begin{bmatrix} e_{z1} \\ e_{z2} \end{bmatrix}$$

$$|\mathbf{e}_x| = |\mathbf{e}_z| = 1, \quad \mathbf{e}_x^T \mathbf{e}_z = 0$$

$$\mathbf{e}_x^T \mathbf{v} = x, \quad \mathbf{e}_z^T \mathbf{v} = z$$

- ❑ **V** defined in local system

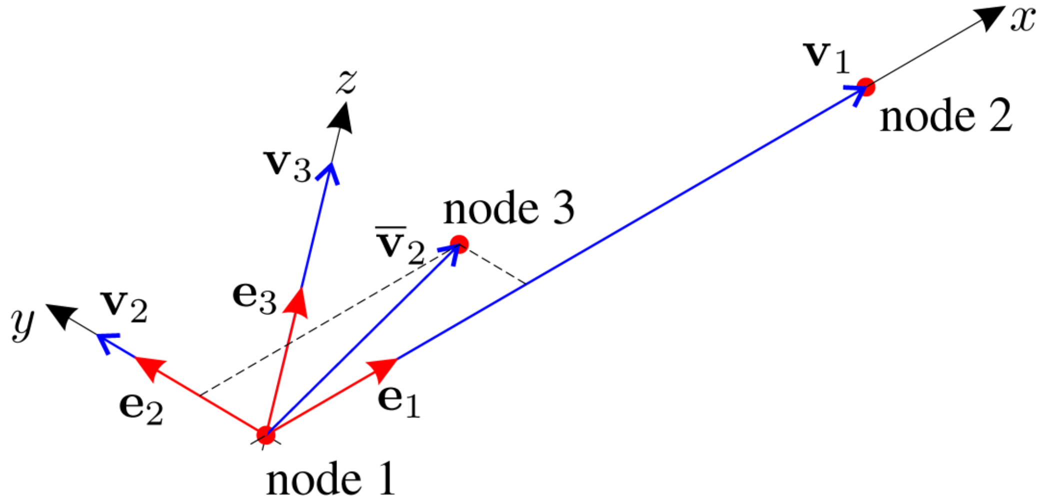
$$\mathbf{v}_l = \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} \mathbf{e}_x^T \\ \mathbf{e}_z^T \end{bmatrix} \mathbf{v}_g = \begin{bmatrix} e_{x1} & e_{x2} \\ e_{z1} & e_{z2} \end{bmatrix} \mathbf{v}_g = \mathbf{T}^T \mathbf{v}_g, \quad \mathbf{T} = \begin{bmatrix} e_{x1} & e_{z1} \\ e_{x2} & e_{z2} \end{bmatrix}$$

- ❑ The transformation matrix is a orthogonal set of unit vectors placed in the columns. This also holds in 3D

Plates and Shells

How do we find the unit vectors describing the xyz-system?

node 1: (x_1, y_1, z_1), node 2: (x_2, y_2, z_2), node 3: (x_3, y_3, z_3)



$$\mathbf{v}_1 = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{bmatrix}, \quad \bar{\mathbf{v}}_2 = \begin{bmatrix} x_3 - x_1 \\ y_3 - y_1 \\ z_3 - z_1 \end{bmatrix}, \quad \mathbf{v}_3 = \mathbf{v}_1 \times \bar{\mathbf{v}}_2, \quad \mathbf{v}_2 = \mathbf{v}_3 \times \mathbf{v}_1$$

$$\mathbf{e}_1 = \frac{\mathbf{v}_1}{|\mathbf{v}_1|}, \quad \mathbf{e}_2 = \frac{\mathbf{v}_2}{|\mathbf{v}_2|}, \quad \mathbf{e}_3 = \frac{\mathbf{v}_3}{|\mathbf{v}_3|}, \quad \mathbf{T} = [\mathbf{e}_1 \ \mathbf{e}_2 \ \mathbf{e}_3] \quad \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} = \mathbf{T}^T \begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix}$$

Exercise: Include the transformation in the program

- ☐ Update transformation.m according to the previous slide.

$$\mathbf{x}_{el} = \mathbf{x}_{eg}\mathbf{T}, \quad \mathbf{x}_{eg} = \begin{bmatrix} \mathbf{0} \\ \mathbf{v}_1^T \\ \mathbf{v}_2^T \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ x_{g2} & y_{g2} & z_{g2} \\ x_{g3} & y_{g3} & z_{g3} \end{bmatrix}, \quad \mathbf{x}_{el} = \begin{bmatrix} 0 & 0 & 0 \\ x_{l2} & 0 & 0 \\ x_{l3} & y_{l3} & 0 \end{bmatrix}$$

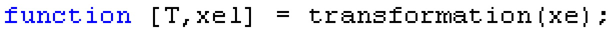
- ☐ Make the full transformation matrix (12x12) from **T** (3x3) in K_3d_beam.m and multiply the local element stiffness matrix with the transformation to obtain the global element stiffness matrix

$$\mathbf{K}_{eg} = \mathbf{T}_g\mathbf{K}_e\mathbf{T}_g^T, \quad \mathbf{T}_g = \begin{bmatrix} \mathbf{T} & & & \mathbf{0} \\ & \mathbf{T} & & \\ & & \mathbf{T} & \\ \mathbf{0} & & & \mathbf{T} \end{bmatrix}$$

  - hint introduce a matrix null = zeros(3,3)
  - The cross product V3xV1 in matlab: `cross(V3,V1);`
  - The transposed: `Tg'`

Exercise 7

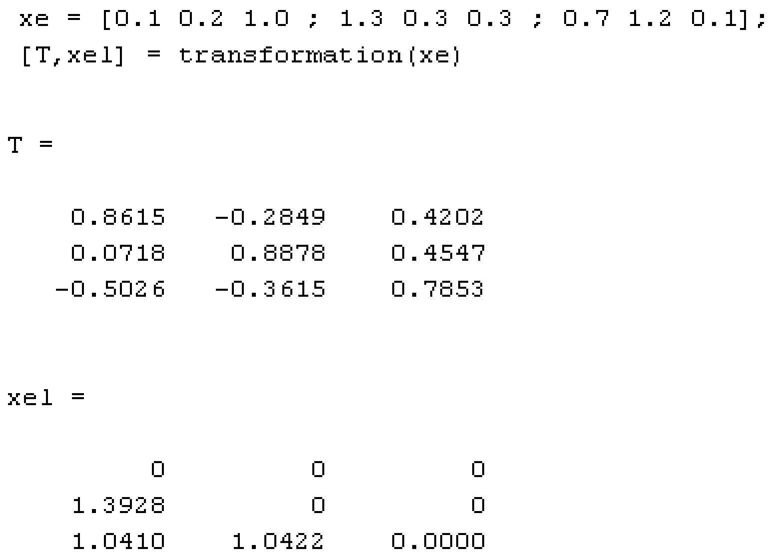
- ❑ Update the transformation function.
  - ◆ input: global element coordinates
  - ◆ output: local element coordinates, 3x3 transformation matrix
  - ◆ Make the full transformation matrix (18x18) and multiply the local element stiffness matrix with the transformation to obtain the global element stiffness matrix



```
function [T,xel] = transformation(xe);
```

$$\mathbf{K}_{EG} = \mathbf{T}_G^T \mathbf{K}_E \mathbf{T}_G, \quad \mathbf{T}_G = \begin{bmatrix} \mathbf{T} & & & & & \\ & \mathbf{T} & & & 0 & \\ & & \mathbf{T} & & & \\ & & & \mathbf{T} & & \\ & 0 & & & \mathbf{T} & \\ & & & & & \mathbf{T} \end{bmatrix}$$

Test the transformation function



```
 xe = [0.1 0.2 1.0 ; 1.3 0.3 0.3 ; 0.7 1.2 0.1];
 [T,xel] = transformation(xe)


T =

    0.8615   -0.2849    0.4202
    0.0718    0.8878    0.4547
   -0.5026   -0.3615    0.7853


xel =

         0         0         0
    1.3928         0         0
    1.0410    1.0422    0.0000
```

Inplane components (Constant-Strain triangle)

- ☐ Kinematic relations

$$\begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \mathbf{L}\bar{\mathbf{u}}$$

- ☐ Equilibrium equation 2D/3D

$$\boxed{\frac{\partial P_x}{\partial x} = 0} \qquad \boxed{\mathbf{L}^T \mathbf{P} = \mathbf{0}} \qquad \mathbf{P} = \begin{bmatrix} P_x \\ P_y \\ P_{xy} \end{bmatrix}$$

$$\mathbf{P} = \int_{-t/2}^{t/2} \boldsymbol{\sigma} dz = \mathbf{D}\mathbf{L}\bar{\mathbf{u}}, \quad \mathbf{D} = \int_{-t/2}^{t/2} \mathbf{E} dz, \quad \mathbf{E} = \frac{E}{(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix}$$

Weak formulation

- ☐ Internal energy

$$\delta\Pi_{int} = \int_\Omega (\delta\varepsilon)^T \mathbf{E}\varepsilon \, d\Omega = \int_\Omega \delta\bar{\mathbf{u}}\mathbf{L}^{\mathbf{T}}\mathbf{E}\mathbf{L}\bar{\mathbf{u}}\,\mathbf{d}\boldsymbol{\Omega}$$

- ☐ Galerkin approach

$$\bar{\mathbf{u}} = \mathbf{N}\mathbf{u} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix}, \quad \delta\bar{\mathbf{u}} = \mathbf{N}\delta\mathbf{u}$$

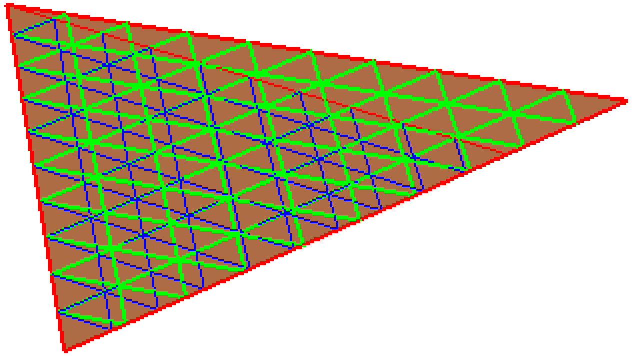
- ☐ Finite Element formulation

$$\mathbf{K}\mathbf{u} = \mathbf{f} \qquad \mathbf{K} = \int_\Omega \mathbf{B}^T\mathbf{E}\mathbf{B}\, d\Omega \quad \mathbf{B} = \mathbf{L}\mathbf{N}$$

Shape functions for the constant strain part

- ☐ The in-plane deformation varies linear over the element, hence,

$$N_1 = L_1, \quad N_2 = L_2, \quad N_3 = L_3$$



- ☐ Derivatives with respect to x and y

$$\begin{bmatrix} \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial L_1}{\partial x} & \dfrac{\partial L_2}{\partial x} & \dfrac{\partial L_3}{\partial x} \\ \dfrac{\partial L_1}{\partial y} & \dfrac{\partial L_2}{\partial y} & \dfrac{\partial L_3}{\partial y} \end{bmatrix} \begin{bmatrix} \dfrac{\partial}{\partial L_1} \\ \dfrac{\partial}{\partial L_2} \\ \dfrac{\partial}{\partial L_3} \end{bmatrix} = \frac{1}{2\Delta} \begin{bmatrix} b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \begin{bmatrix} \dfrac{\partial}{\partial L_1} \\ \dfrac{\partial}{\partial L_2} \\ \dfrac{\partial}{\partial L_3} \end{bmatrix}$$

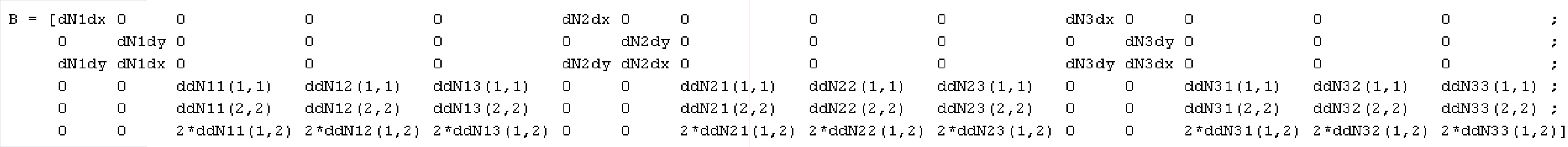
- ☐ Notice that the derivative of the shape functions give constant values across the element, i.e. the strain components are constant over the element, hence, constant strain triangle.

Exercise 8

- ☐ Include the constant strain part in the shape functions
  - N should be organized as follows

```
N = [N1 0  0   0   0   N2 0  0   0   0   N3 0  0   0   0   ;
     0  N1 0   0   0   0  N2 0   0   0   0  N3 0   0   0   ;
     0  0  N11 N12 N13 0  0  N21 N22 N23 0  0  N31 N32 N33 ];
```

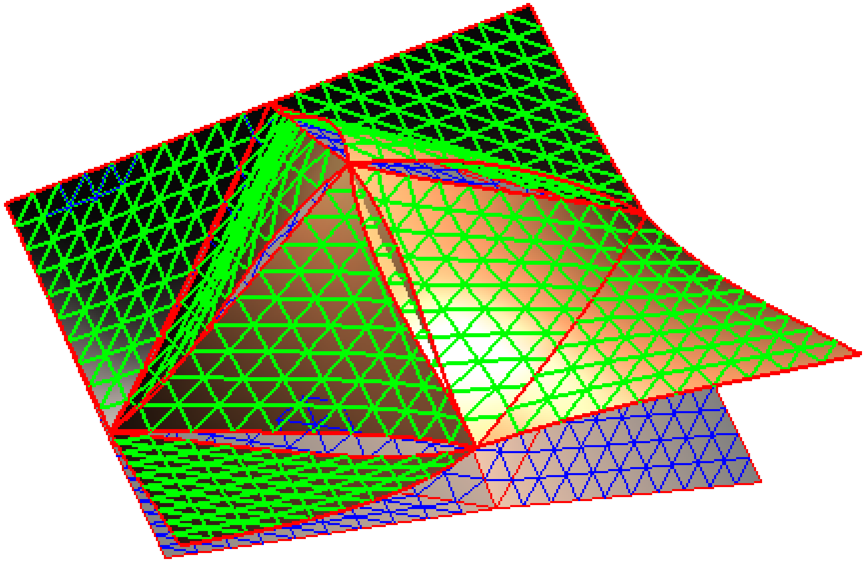
  - B should be organized as follows



```
B = [dN1dx 0     0             0             0             dN2dx 0     0             0             0             dN3dx 0     0             0             0             ;
     0     dN1dy 0             0             0             0     dN2dy 0             0             0             0     dN3dy 0             0             0             ;
     dN1dy dN1dx 0             0             0             dN2dy dN2dx 0             0             0             dN3dy dN3dx 0             0             0             ;
     0     0     ddN11(1,1)    ddN12(1,1)    ddN13(1,1)    0     0     ddN21(1,1)    ddN22(1,1)    ddN23(1,1)    0     0     ddN31(1,1)    ddN32(1,1)    ddN33(1,1)    ;
     0     0     ddN11(2,2)    ddN12(2,2)    ddN13(2,2)    0     0     ddN21(2,2)    ddN22(2,2)    ddN23(2,2)    0     0     ddN31(2,2)    ddN32(2,2)    ddN33(2,2)    ;
     0     0     2*ddN11(1,2)  2*ddN12(1,2)  2*ddN13(1,2)  0     0     2*ddN21(1,2)  2*ddN22(1,2)  2*ddN23(1,2)  0     0     2*ddN31(1,2)  2*ddN32(1,2)  2*ddN33(1,2)]
```

Exercise 9

- ☐ Try your new plate elements in various configurations (2- and 3-dimensional)
- ☐ Remove the bugs (if you have any!)



Laminted plate with orthotropic material

Constitutive models in linear elasticity, 3D

- stress vector

$$\boldsymbol{\sigma} = [\sigma_{11}\ \sigma_{22}\ \sigma_{33}\ \tau_{12}\ \tau_{23}\ \tau_{31}]^T$$

- Strain vector

$$\boldsymbol{\varepsilon} = [\varepsilon_{11}\ \varepsilon_{22}\ \varepsilon_{33}\ \gamma_{12}\ \gamma_{23}\ \gamma_{31}]^T$$

- Isotropic behaviour in 3D

Flexibility matrix

$$\boldsymbol{\varepsilon} = \mathbf{C}\boldsymbol{\sigma}, \quad \mathbf{C} = \begin{bmatrix} \frac{1}{E} & -\frac{\nu}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\ -\frac{\nu}{E} & \frac{1}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\ -\frac{\nu}{E} & -\frac{\nu}{E} & \frac{1}{E} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G} \end{bmatrix}$$

Shear modulus

$$G = \frac{E}{2(1+\nu)}$$

Plates and Shells

$$\boldsymbol{\varepsilon} = \mathbf{C}\boldsymbol{\sigma}, \quad \mathbf{C} = \begin{bmatrix} \frac{1}{E} & -\frac{\nu}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\ -\frac{\nu}{E} & \frac{1}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\ -\frac{\nu}{E} & -\frac{\nu}{E} & \frac{1}{E} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G} \end{bmatrix} \quad G = \frac{E}{2(1+\nu)}$$

$$\mathbf{E} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}$$

Isotropic behaviour of plate or shell

- Plane stress $\sigma_{33} = 0$
- Normal strain ε_{33} is disregarded

$$\begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \gamma_{12} \\ \gamma_{23} \\ \gamma_{31} \end{bmatrix} = \begin{bmatrix} \frac{1}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\ -\frac{\nu}{E} & \frac{1}{E} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{G} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G} \end{bmatrix} \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \tau_{12} \\ \tau_{23} \\ \tau_{31} \end{bmatrix}$$

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \tau_{12} \\ \tau_{23} \\ \tau_{31} \end{bmatrix} = \begin{bmatrix} \frac{E}{1-\nu^2} & -\frac{E\nu}{1-\nu^2} & 0 & 0 & 0 \\ -\frac{E\nu}{1-\nu^2} & \frac{E}{1-\nu^2} & 0 & 0 & 0 \\ 0 & 0 & G & 0 & 0 \\ 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & G \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \gamma_{12} \\ \gamma_{23} \\ \gamma_{31} \end{bmatrix}$$

Orthotropic behaviour of plate or shell

- Different Young’s moduli $E_1 \neq E_2$
- Different Poisson’s ratios $\nu_{12} \neq \nu_{21}$
- Shear moduli are unrelated
  - Need not be related to $E_1, E_2, \nu_{12}, \nu_{21}$

Orthotropic behaviour of plate or shell

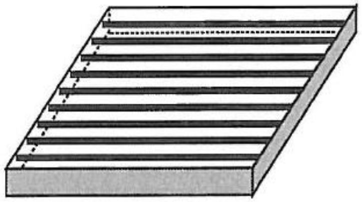
$$\begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \gamma_{12} \\ \gamma_{23} \\ \gamma_{31} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_1} & -\frac{\nu_{21}}{E_2} & 0 & 0 & 0 \\ -\frac{\nu_{12}}{E_1} & \frac{1}{E_2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{G_{12}} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{23}} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{31}} \end{bmatrix} \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \tau_{12} \\ \tau_{23} \\ \tau_{31} \end{bmatrix}$$

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \tau_{12} \\ \tau_{23} \\ \tau_{31} \end{bmatrix} = \begin{bmatrix} \frac{E_1}{1-\nu_{12}\nu_{21}} & -\frac{E_1\nu_{21}}{1-\nu_{12}\nu_{21}} & 0 & 0 & 0 \\ -\frac{E_2\nu_{12}}{1-\nu_{12}\nu_{21}} & \frac{E_2}{1-\nu_{12}\nu_{21}} & 0 & 0 & 0 \\ 0 & 0 & G_{12} & 0 & 0 \\ 0 & 0 & 0 & \kappa G_{23} & 0 \\ 0 & 0 & 0 & 0 & \kappa G_{31} \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \gamma_{12} \\ \gamma_{23} \\ \gamma_{31} \end{bmatrix}$$

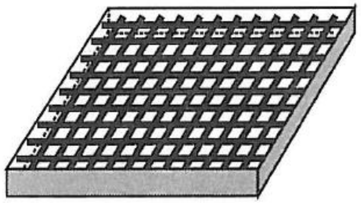
- The shear stresses τ_{23}, τ_{31} are uniformly distributede in the model. In reality they are closer to a parabolic distribution. The stiffness related to these values are overpredicted by a factor $\frac{1}{\kappa}$ (κ=5/6 rectangular cross sections)

Composite Fibre materials

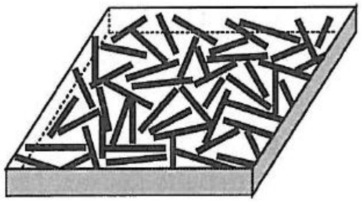
- ☐ Matrix material (basis material, e.g. concrete)
- ☐ Fibre material (reinforcement, e.g. steel)



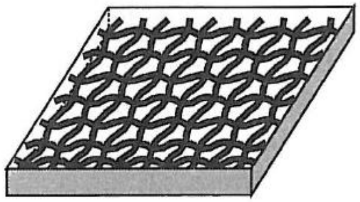
(a) Unidirectional



(b) Bi–directional



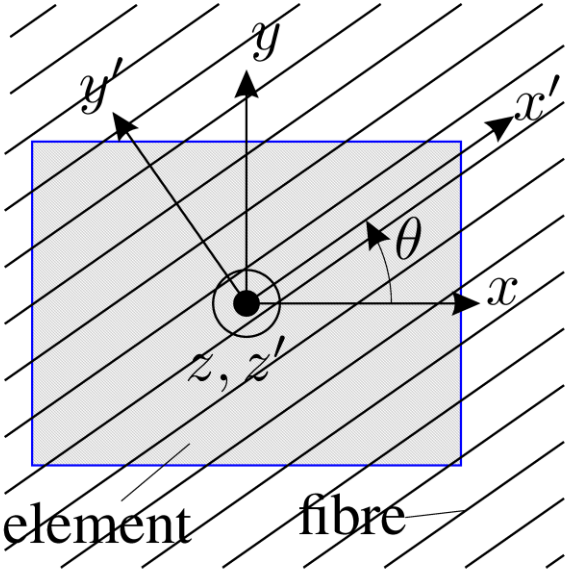
(c) Discontinuous fiber



(d) Woven

Fiber directions and element system

- ☐ We would like to calculate the element stiffness matrix in an element coordinate system (x,y)
- ☐ The material properties are often given in the fiber directions (x’,y’)



- ☐ Transformation

$$\mathbf{u}' = \mathbf{L}\mathbf{u}, \quad \mathbf{u} = \mathbf{L}^{-1}\mathbf{u} = \mathbf{L}^{T}\mathbf{u}, \quad \mathbf{L} = [L_{ij}] = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Transformation of tensors

$$u'_i = L_{ij} u_j \quad \Leftrightarrow \quad u_i = L_{ji} u'_j$$

$$\varepsilon'_{ij} = L_{ik} \varepsilon_{kl} L_{jl} \quad \Leftrightarrow \quad \varepsilon_{ij} = L_{ki} \varepsilon'_{kl} L_{jl}$$

$$\sigma'_{ij} = L_{ik} \sigma_{kl} L_{jl} \quad \Leftrightarrow \quad \sigma_{ij} = L_{ki} \sigma'_{kl} L_{jl}$$

$$\sigma'_{ij} = E'_{ijkl} \varepsilon'_{kl} = E'_{ijkl} L_{km} \varepsilon_{mn} L_{ln}$$

$$\sigma_{ij} = L_{oi} E'_{opkl} L_{km} \varepsilon_{mn} L_{ln} L_{pj} = L_{oi} L_{pj} E'_{opkl} L_{km} L_{ln} \varepsilon_{mn} = E_{ijmn} \varepsilon_{mn}$$

$$E_{ijkl} = L_{oi} L_{pj} E'_{opmn} L_{mk} L_{nl}$$

Matrix vector form of the transformation

$$\boldsymbol{\sigma} = \mathbf{T}\boldsymbol{\sigma}' = \mathbf{T}\mathbf{E}'\boldsymbol{\varepsilon}' = \mathbf{T}\mathbf{E}'\mathbf{T}^T\boldsymbol{\varepsilon}$$

$$\mathbf{T} = \begin{bmatrix} \cos^2\theta & \sin^2\theta & -2\sin\theta\cos\theta & 0 & 0 \\ \sin^2\theta & \cos^2\theta & 2\sin\theta\cos\theta & 0 & 0 \\ \sin\theta\cos\theta & -\sin\theta\cos\theta & \cos^2\theta - \sin^2\theta & 0 & 0 \\ 0 & 0 & 0 & \cos\theta & \sin\theta \\ 0 & 0 & 0 & \sin\theta & \cos\theta \end{bmatrix}$$

- ☐ **T** holds the components $L_{ij}L_{kl}$, i.e.

$$\mathbf{E} = \mathbf{T}\mathbf{E}'\mathbf{T}^T \quad \sim \quad E_{ijkl} = L_{oi}L_{pj}E'_{opmn}L_{mk}L_{nl}$$

Rotation of material properties, Kirchhoff plates, see slide 18

$$\begin{bmatrix}\sigma_{11}\\ \sigma_{22}\\ \tau_{12}\end{bmatrix} = \begin{bmatrix}\frac{E_1}{1-\nu_{12}\nu_{21}} & -\frac{E_1\nu_{21}}{1-\nu_{12}\nu_{21}} & 0\\ -\frac{E_2\nu_{12}}{1-\nu_{12}\nu_{21}} & \frac{E_2}{1-\nu_{12}\nu_{21}} & 0\\ 0 & 0 & G_{12}\end{bmatrix}\begin{bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \gamma_{12}\end{bmatrix}$$

$\mathbf{E}'$

$$\mathbf{E} = \mathbf{T}\mathbf{E}'\mathbf{T}^T \qquad \mathbf{T} = \begin{bmatrix}\cos^2\theta & \sin^2\theta & -2\sin\theta\cos\theta\\ \sin^2\theta & \cos^2\theta & 2\sin\theta\cos\theta\\ \sin\theta\cos\theta & -\sin\theta\cos\theta & \cos^2\theta-\sin^2\theta\end{bmatrix}$$

- ☐ Organizing the constitutive matrix due to symmetry

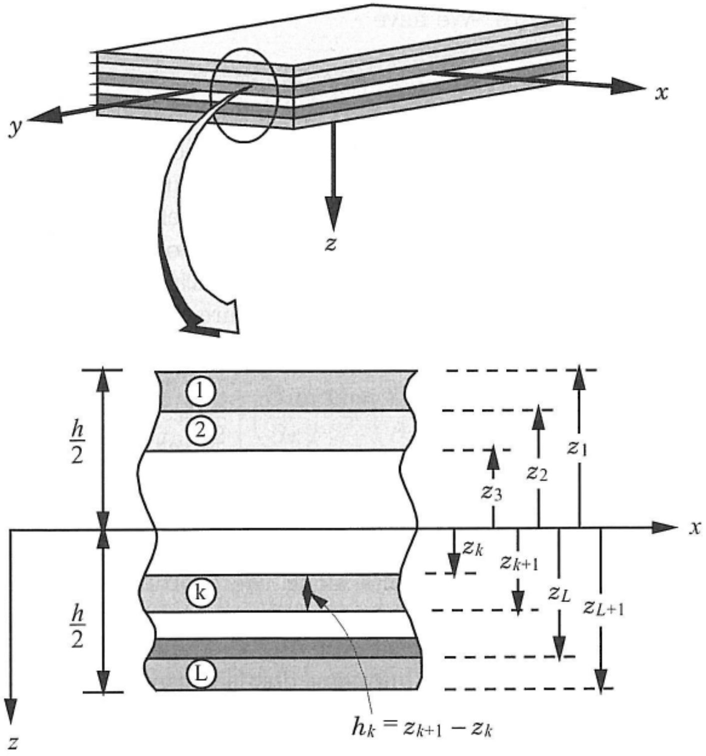
$$\mathbf{Q} = [E_{11}\ E_{12}\ E_{22}\ E_{13}\ E_{23}\ E_{33}]$$

Exercise: Identify the rotation of material properties in the program

☐ Look through the program and identify the function where the material properties are rotated, and the constitutive relation is determined

Laminated plates

- ☐ A laminate plate consists of a number of material layers – so-called lamina. The material in each lamina is typically orthotropic due to fiber reinforcement.
  - ◆ Glass fibre, reinforced concrete, sandwich panel (in e.g. aeroplanes)
- ☐ I.e. we need to specify:
  - ◆ Orthotropic material properties for each lamina, (E_1,E_2, ν_{12},G_{12},G_{23},G_{31})
  - ◆ Thickness for each lamina
  - ◆ The setup (numbering of lamina)

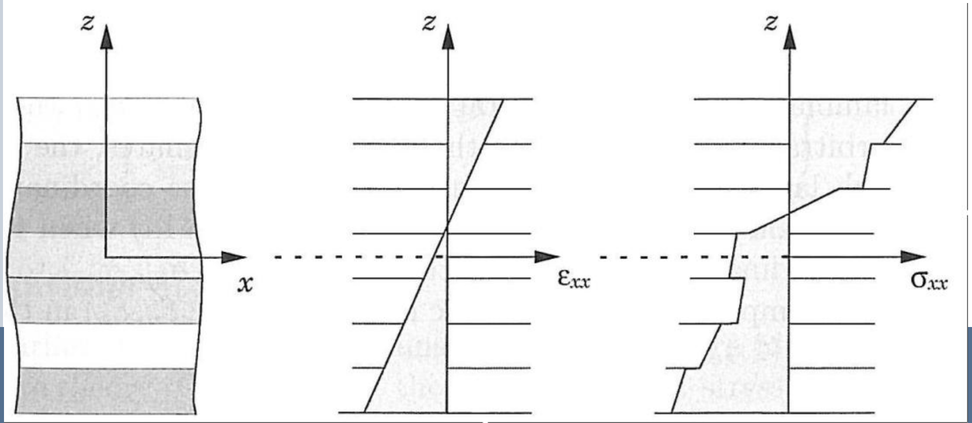


Constitutive relation for laminated plates

- ☐ The constitutive matrix **D** originates from integrating the material properties over stiffness. This stems from the definition of forces and moments
- ☐ See Slide 19 for moment definition and slide 48 for forces

$$M_x = -\int_{-t/2}^{t/2} z\sigma_x dz, \quad M_y = -\int_{-t/2}^{t/2} z\sigma_y dz, \quad M_{xy} = -\int_{-t/2}^{t/2} z\tau_{xy} dz$$

$$P_x = \int_{-t/2}^{t/2} \sigma_x dz, \quad P_y = \int_{-t/2}^{t/2} \sigma_y dz, \quad P_{xy} = \int_{-t/2}^{t/2} \tau_{xy} dz$$



- ❑ Rotated constitutive matrix for the *i*'th lamina is determined

$$\mathbf{E}_i = \mathbf{T}\mathbf{E}_i'\mathbf{T}^T$$

- ❑ Constitutive relation for the *i*'th lamina

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix}_i = \mathbf{E}_i \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix}_i = \mathbf{E}_i \left(\begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{bmatrix} - z \begin{bmatrix} \frac{\partial^2 w}{\partial x^2} \\ \frac{\partial^2 w}{\partial y^2} \\ 2\frac{\partial^2 w}{\partial x \partial y} \end{bmatrix} \right)_i \quad \Rightarrow$$

In-plane deformation

bending deformation

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix}_i = \begin{bmatrix} \mathbf{E}_i & -z\mathbf{E}_i \end{bmatrix} \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial^2 w}{\partial x^2} \\ \frac{\partial^2 w}{\partial y^2} \\ 2\frac{\partial^2 w}{\partial x \partial y} \end{bmatrix}_i$$

Plates and Shells

- ☐ Inserting the stress components in the forces and moments provides

$$\begin{bmatrix} P_x \\ P_y \\ P_{xy} \\ M_x \\ M_y \\ M_{xy} \end{bmatrix} = \begin{bmatrix} \mathbf{AA} & -\mathbf{BB} \\ -\underset{\text{6x6}}{\mathbf{BB}} & \mathbf{DD} \end{bmatrix} \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial^2 w}{\partial x^2} \\ \frac{\partial^2 w}{\partial y^2} \\ 2\frac{\partial^2 w}{\partial x \partial y} \end{bmatrix}$$

$$\mathbf{AA} = \sum_{i=1}^{N} \int_{z_i}^{z_{i+1}} \mathbf{E}_i dz, \quad \mathbf{BB} = \sum_{i=1}^{N} \int_{z_i}^{z_{i+1}} z\mathbf{E}_i dz, \quad \mathbf{DD} = \sum_{i=1}^{N} \int_{z_i}^{z_{i+1}} z^2\mathbf{E}_i dz$$

- ☐ The integrals can be evaluated in advance of the element stiffness matrix (done for the plates)
- ☐ Or during the calculation of the element stiffness matrix in each Gauss point by e.g. Gauss quadrature (done for degenerated shell elements)

Plates and Shells

- ☐ Solving the integrals, assuming constant **E** for each lamina

$$\mathbf{AA} = \sum_{i=1}^{N} \int_{z_i}^{z_{i+1}} \mathbf{E}_i dz, \quad \mathbf{BB} = \sum_{i=1}^{N} \int_{z_i}^{z_{i+1}} z\mathbf{E}_i dz, \quad \mathbf{DD} = \sum_{i=1}^{N} \int_{z_i}^{z_{i+1}} z^2\mathbf{E}_i dz$$

$$\mathbf{AA} = \sum_{i=1}^{N} (z_{i+1} - z_i)\mathbf{E}_i dz, \quad \mathbf{BB} = \sum_{i=1}^{N} \frac{1}{2}(z_{i+1}^2 - z_i^2)\mathbf{E}_i dz,$$

$$\mathbf{DD} = \sum_{i=1}^{N} \frac{1}{3}(z_{i+1}^3 - z_i^2) z^3 \mathbf{E}_i dz$$

$$\mathbf{CC} = \begin{bmatrix} \mathbf{AA} & -\mathbf{BB} \\ -\mathbf{BB} & \mathbf{DD} \end{bmatrix}$$

$$\mathbf{K} = \int_{\Omega} \underset{\text{18x6}}{\mathbf{B}^T} \underset{\text{6x6}}{\mathbf{CC}} \underset{\text{6x18}}{\mathbf{B}} \, d\Omega$$

Exercise: Identify the integral over plate height of the constitutive relations for each lamina

- ☐ Create a laminated plate with three lamina with height 0.05m, 0.07m and 0.08m respectively. The fibre directions should be 0°, 20° and 35°.
- ☐ $E_1 = 2e8$, $E_2 = 1e8$, $\nu_{12} = 0.3$ and $G_{12} = 1e8$, $G_{23}=G_{31}=0$
- ☐ Make a patch test
- ☐ Explain the steps in evaluating a laminated plate stiffness

What did you learn to day?

- steps in the Finite Element Method (FEM)
- Theory of a Kirchhoff plate element
  - Strong formulation
  - Weak formulation
- Implementing a 3-node Kirchhoff plate element into the Matlab program
- Area coordinates
  - Gauss quadrature using area coordinates
- Shape functions for 3-node element
  - N- and B-matrix for 3-node Kirchhoff plate element
- Transformation of degrees of freedom and stiffness matrix
- How to include the inplane constant-strain element into the formulation
- Laminated plates of orthotropic material

Thank you for your attention

Solution exercise 1

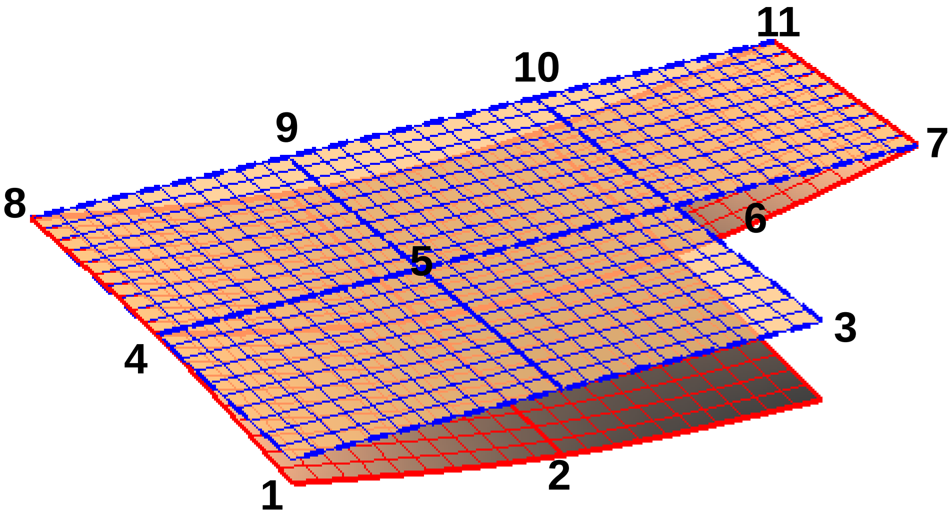
- ❑ Pre
  - ◆ Materials
  - ◆ Node coordinates
  - ◆ Topology (how nodes are connected to the elements)
  - ◆ Boundary conditions (loads, supports)
  - ◆ Global numbering of DOF
- ❑ Analysis
  - ◆ Konstitutive model (relation between strains and stresses)
  - ◆ Stiffness matrix and mass matrix for each element
    - ❑ Define shape functions
    - ❑ Integration over elements (stiffness, mass), e.g. by quadrature
    - ❑ Rotate the stiffness into a global system
  - ◆ Assemble the global stiffness matrix
  - ◆ Remove support DOF from the equations
  - ◆ Solve the system equation (**Ku**=**f**) for the DOF (translation and rotation)

Solution exercise 1

- Post
  - Determine the displacement field across elements
  - Determine strain components across elements
  - Determine stress components across elements
  - Plot the results

Solution exercise 2

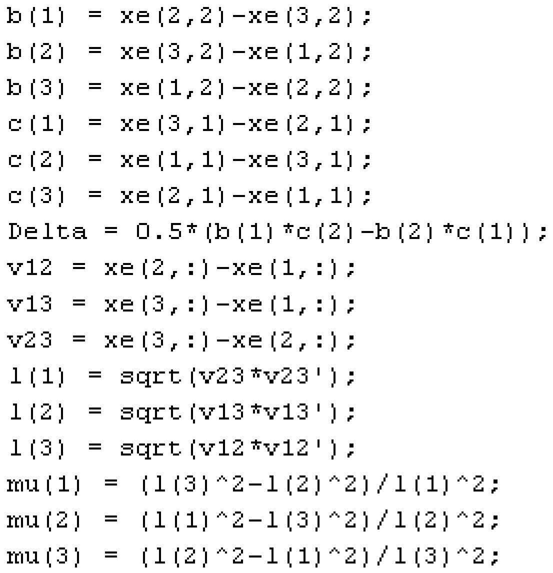
```
Xcoord = [ 0 0 0 ; 1 0 0 ; 2 0 0 ;
           0 1 0 ; 1 1 0 ; 2 1 0; 3 1 0;
           0 2 0 ; 1 2 0 ; 2 2 0; 3 2 0 ]  ;

 Top = [ etype  1  1  2  5  4
         etype  1  2  3  6  5
         etype  1  4  5  9  8
         etype  1  5  6  10  9
         etype  1  6  7  11  10] ;

 BC = [    4    1    0    0
           4    2    0    0
           4    3    0    0
           8    1    0    0
           8    2    0    0
           8    3    0    0
           7    1    0    0
           7    2    0    0
           7    3    0    0
           11    1    0    0
           11    2    0    0
           11    3    0    0
           5    3    1    -5000 ] ;
```



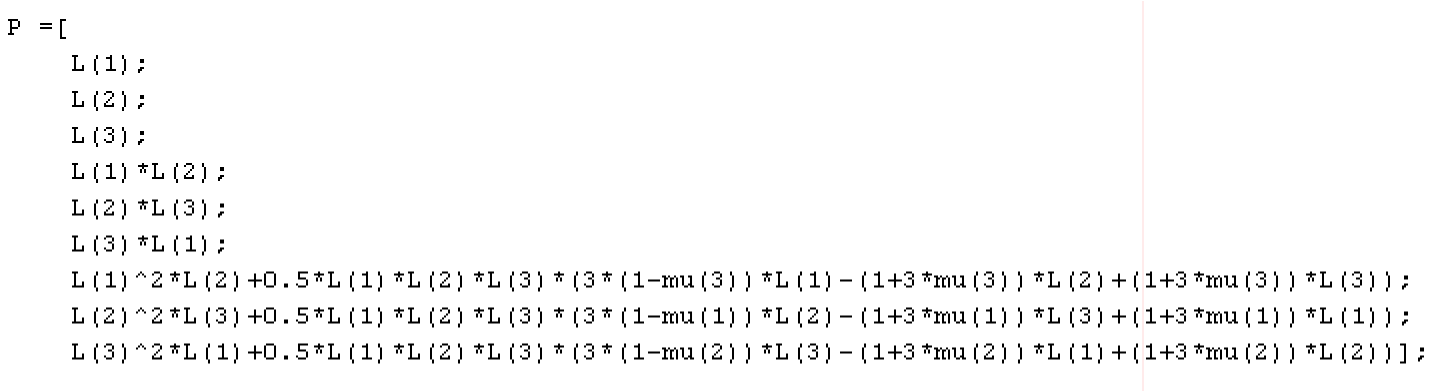
Solution exercise 3

- Topology matrix defines the element from the nodes. Top (main)
- Global numbering of DOF. en=3 nNodeDof = [6 6 6] (elemtype)
- Stiffness matrix for a 3-node element
    - Define Gauss-points
    - Determine the constitutive matrix (the same as for 4-node element)
    - Loop over Gauss-points
        - determine shape functions
        - determine stiffness contribution from the Gauss-point
        - add the contribution to the total element stiffness matrix
- Plot function (already made)

Solution exercise 4

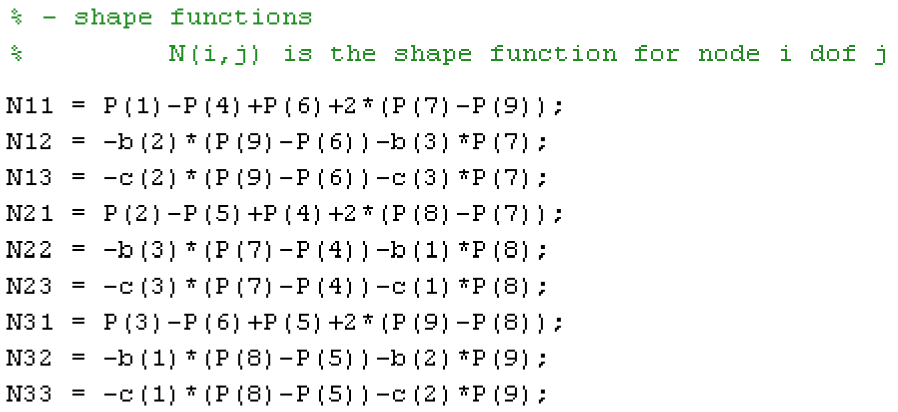


```
b(1)  = xe(2,2)-xe(3,2);
b(2)  = xe(3,2)-xe(1,2);
b(3)  = xe(1,2)-xe(2,2);
c(1)  = xe(3,1)-xe(2,1);
c(2)  = xe(1,1)-xe(3,1);
c(3)  = xe(2,1)-xe(1,1);
Delta = 0.5*(b(1)*c(2)-b(2)*c(1));
v12 = xe(2,:)-xe(1,:);
v13 = xe(3,:)-xe(1,:);
v23 = xe(3,:)-xe(2,:);
l(1)  = sqrt(v23*v23');
l(2)  = sqrt(v13*v13');
l(3)  = sqrt(v12*v12');
mu(1)  =  (l(3)^2-l(2)^2)/l(1)^2;
mu(2)  =  (l(1)^2-l(3)^2)/l(2)^2;
mu(3)  =  (l(2)^2-l(1)^2)/l(3)^2;
```

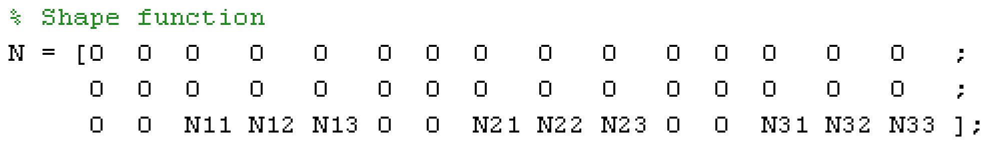


```
P =[
    L(1);
    L(2);
    L(3);
    L(1)*L(2);
    L(2)*L(3);
    L(3)*L(1);
    L(1)^2*L(2)+0.5*L(1)*L(2)*L(3)*(3*(1-mu(3))*L(1)-(1+3*mu(3))*L(2)+(1+3*mu(3))*L(3));
    L(2)^2*L(3)+0.5*L(1)*L(2)*L(3)*(3*(1-mu(1))*L(2)-(1+3*mu(1))*L(3)+(1+3*mu(1))*L(1));
    L(3)^2*L(1)+0.5*L(1)*L(2)*L(3)*(3*(1-mu(2))*L(3)-(1+3*mu(2))*L(1)+(1+3*mu(2))*L(2))];
```

Solution exercise 4, continued

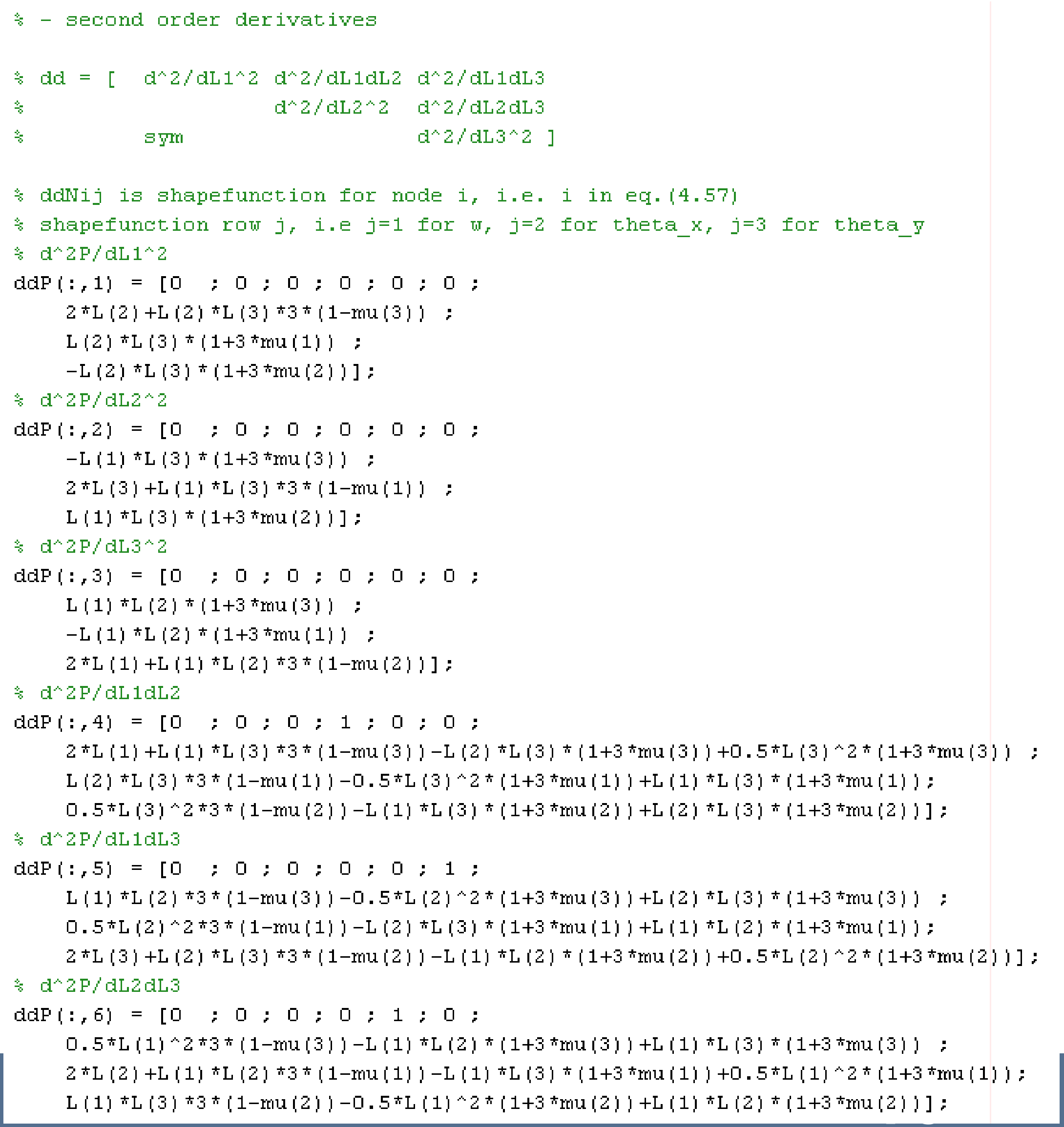


```
% - shape functions
%         N(i,j) is the shape function for node i dof j

N11 = P(1)-P(4)+P(6)+2*(P(7)-P(9));
N12 = -b(2)*(P(9)-P(6))-b(3)*P(7);
N13 = -c(2)*(P(9)-P(6))-c(3)*P(7);
N21 = P(2)-P(5)+P(4)+2*(P(8)-P(7));
N22 = -b(3)*(P(7)-P(4))-b(1)*P(8);
N23 = -c(3)*(P(7)-P(4))-c(1)*P(8);
N31 = P(3)-P(6)+P(5)+2*(P(9)-P(8));
N32 = -b(1)*(P(8)-P(5))-b(2)*P(9);
N33 = -c(1)*(P(8)-P(5))-c(2)*P(9);
```



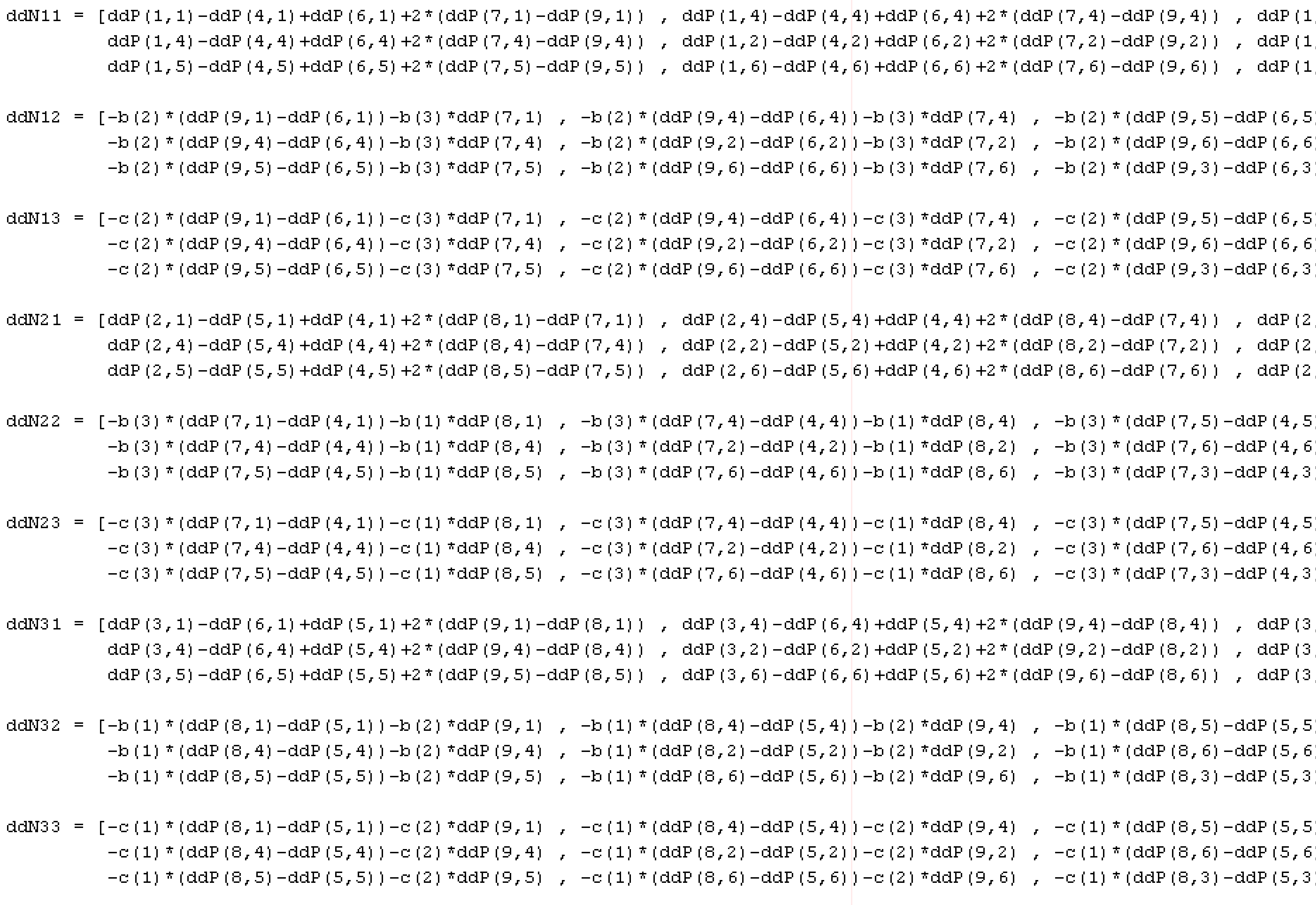
```
% Shape function
N = [0  0  0   0   0   0  0  0   0   0   0  0  0   0   0   ;
     0  0  0   0   0   0  0  0   0   0   0  0  0   0   0   ;
     0  0  N11 N12 N13 0  0  N21 N22 N23 0  0  N31 N32 N33 ];
```

Solution exercise 5



```
% - second order derivatives

% dd = [  d^2/dL1^2 d^2/dL1dL2 d^2/dL1dL3
%                   d^2/dL2^2  d^2/dL2dL3
%         sym                  d^2/dL3^2 ]

% ddNij is shapefunction for node i, i.e. i in eq.(4.57)
% shapefunction row j, i.e j=1 for w, j=2 for theta_x, j=3 for theta_y
% d^2P/dL1^2
ddP(:,1) = [0  ; 0 ; 0 ; 0 ; 0 ; 0 ;
    2*L(2)+L(2)*L(3)*3*(1-mu(3)) ;
    L(2)*L(3)*(1+3*mu(1)) ;
    -L(2)*L(3)*(1+3*mu(2))];
% d^2P/dL2^2
ddP(:,2) = [0  ; 0 ; 0 ; 0 ; 0 ; 0 ;
    -L(1)*L(3)*(1+3*mu(3)) ;
    2*L(3)+L(1)*L(3)*3*(1-mu(1)) ;
    L(1)*L(3)*(1+3*mu(2))];
% d^2P/dL3^2
ddP(:,3) = [0  ; 0 ; 0 ; 0 ; 0 ; 0 ;
    L(1)*L(2)*(1+3*mu(3)) ;
    -L(1)*L(2)*(1+3*mu(1)) ;
    2*L(1)+L(1)*L(2)*3*(1-mu(2))];
% d^2P/dL1dL2
ddP(:,4) = [0  ; 0 ; 0 ; 1 ; 0 ; 0 ;
    2*L(1)+L(1)*L(3)*3*(1-mu(3))-L(2)*L(3)*(1+3*mu(3))+0.5*L(3)^2*(1+3*mu(3)) ;
    L(2)*L(3)*3*(1-mu(1))-0.5*L(3)^2*(1+3*mu(1))+L(1)*L(3)*(1+3*mu(1));
    0.5*L(3)^2*3*(1-mu(2))-L(1)*L(3)*(1+3*mu(2))+L(2)*L(3)*(1+3*mu(2))];
% d^2P/dL1dL3
ddP(:,5) = [0  ; 0 ; 0 ; 0 ; 0 ; 1 ;
    L(1)*L(2)*3*(1-mu(3))-0.5*L(2)^2*(1+3*mu(3))+L(2)*L(3)*(1+3*mu(3)) ;
    0.5*L(2)^2*3*(1-mu(1))-L(2)*L(3)*(1+3*mu(1))+L(1)*L(2)*(1+3*mu(1));
    2*L(3)+L(2)*L(3)*3*(1-mu(2))-L(1)*L(2)*(1+3*mu(2))+0.5*L(2)^2*(1+3*mu(2))];
% d^2P/dL2dL3
ddP(:,6) = [0  ; 0 ; 0 ; 0 ; 1 ; 0 ;
    0.5*L(1)^2*3*(1-mu(3))-L(1)*L(2)*(1+3*mu(3))+L(1)*L(3)*(1+3*mu(3)) ;
    2*L(2)+L(1)*L(2)*3*(1-mu(1))-L(1)*L(3)*(1+3*mu(1))+0.5*L(1)^2*(1+3*mu(1));
    L(1)*L(3)*3*(1-mu(2))-0.5*L(1)^2*(1+3*mu(2))+L(1)*L(2)*(1+3*mu(2))];
```

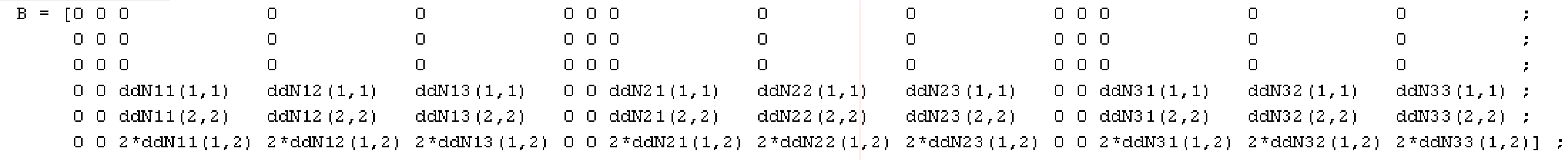
Plates and Shells



```
ddN11 = [ddP(1,1)-ddP(4,1)+ddP(6,1)+2*(ddP(7,1)-ddP(9,1)) , ddP(1,4)-ddP(4,4)+ddP(6,4)+2*(ddP(7,4)-ddP(9,4)) , ddP(1
          ddP(1,4)-ddP(4,4)+ddP(6,4)+2*(ddP(7,4)-ddP(9,4)) , ddP(1,2)-ddP(4,2)+ddP(6,2)+2*(ddP(7,2)-ddP(9,2)) , ddP(1
          ddP(1,5)-ddP(4,5)+ddP(6,5)+2*(ddP(7,5)-ddP(9,5)) , ddP(1,6)-ddP(4,6)+ddP(6,6)+2*(ddP(7,6)-ddP(9,6)) , ddP(1

ddN12 = [-b(2)*(ddP(9,1)-ddP(6,1))-b(3)*ddP(7,1) , -b(2)*(ddP(9,4)-ddP(6,4))-b(3)*ddP(7,4) , -b(2)*(ddP(9,5)-ddP(6,5
          -b(2)*(ddP(9,4)-ddP(6,4))-b(3)*ddP(7,4) , -b(2)*(ddP(9,2)-ddP(6,2))-b(3)*ddP(7,2) , -b(2)*(ddP(9,6)-ddP(6,6
          -b(2)*(ddP(9,5)-ddP(6,5))-b(3)*ddP(7,5) , -b(2)*(ddP(9,6)-ddP(6,6))-b(3)*ddP(7,6) , -b(2)*(ddP(9,3)-ddP(6,3

ddN13 = [-c(2)*(ddP(9,1)-ddP(6,1))-c(3)*ddP(7,1) , -c(2)*(ddP(9,4)-ddP(6,4))-c(3)*ddP(7,4) , -c(2)*(ddP(9,5)-ddP(6,5
          -c(2)*(ddP(9,4)-ddP(6,4))-c(3)*ddP(7,4) , -c(2)*(ddP(9,2)-ddP(6,2))-c(3)*ddP(7,2) , -c(2)*(ddP(9,6)-ddP(6,6
          -c(2)*(ddP(9,5)-ddP(6,5))-c(3)*ddP(7,5) , -c(2)*(ddP(9,6)-ddP(6,6))-c(3)*ddP(7,6) , -c(2)*(ddP(9,3)-ddP(6,3

ddN21 = [ddP(2,1)-ddP(5,1)+ddP(4,1)+2*(ddP(8,1)-ddP(7,1)) , ddP(2,4)-ddP(5,4)+ddP(4,4)+2*(ddP(8,4)-ddP(7,4)) , ddP(2
          ddP(2,4)-ddP(5,4)+ddP(4,4)+2*(ddP(8,4)-ddP(7,4)) , ddP(2,2)-ddP(5,2)+ddP(4,2)+2*(ddP(8,2)-ddP(7,2)) , ddP(2
          ddP(2,5)-ddP(5,5)+ddP(4,5)+2*(ddP(8,5)-ddP(7,5)) , ddP(2,6)-ddP(5,6)+ddP(4,6)+2*(ddP(8,6)-ddP(7,6)) , ddP(2

ddN22 = [-b(3)*(ddP(7,1)-ddP(4,1))-b(1)*ddP(8,1) , -b(3)*(ddP(7,4)-ddP(4,4))-b(1)*ddP(8,4) , -b(3)*(ddP(7,5)-ddP(4,5
          -b(3)*(ddP(7,4)-ddP(4,4))-b(1)*ddP(8,4) , -b(3)*(ddP(7,2)-ddP(4,2))-b(1)*ddP(8,2) , -b(3)*(ddP(7,6)-ddP(4,6
          -b(3)*(ddP(7,5)-ddP(4,5))-b(1)*ddP(8,5) , -b(3)*(ddP(7,6)-ddP(4,6))-b(1)*ddP(8,6) , -b(3)*(ddP(7,3)-ddP(4,3

ddN23 = [-c(3)*(ddP(7,1)-ddP(4,1))-c(1)*ddP(8,1) , -c(3)*(ddP(7,4)-ddP(4,4))-c(1)*ddP(8,4) , -c(3)*(ddP(7,5)-ddP(4,5
          -c(3)*(ddP(7,4)-ddP(4,4))-c(1)*ddP(8,4) , -c(3)*(ddP(7,2)-ddP(4,2))-c(1)*ddP(8,2) , -c(3)*(ddP(7,6)-ddP(4,6
          -c(3)*(ddP(7,5)-ddP(4,5))-c(1)*ddP(8,5) , -c(3)*(ddP(7,6)-ddP(4,6))-c(1)*ddP(8,6) , -c(3)*(ddP(7,3)-ddP(4,3

ddN31 = [ddP(3,1)-ddP(6,1)+ddP(5,1)+2*(ddP(9,1)-ddP(8,1)) , ddP(3,4)-ddP(6,4)+ddP(5,4)+2*(ddP(9,4)-ddP(8,4)) , ddP(3
          ddP(3,4)-ddP(6,4)+ddP(5,4)+2*(ddP(9,4)-ddP(8,4)) , ddP(3,2)-ddP(6,2)+ddP(5,2)+2*(ddP(9,2)-ddP(8,2)) , ddP(3
          ddP(3,5)-ddP(6,5)+ddP(5,5)+2*(ddP(9,5)-ddP(8,5)) , ddP(3,6)-ddP(6,6)+ddP(5,6)+2*(ddP(9,6)-ddP(8,6)) , ddP(3

ddN32 = [-b(1)*(ddP(8,1)-ddP(5,1))-b(2)*ddP(9,1) , -b(1)*(ddP(8,4)-ddP(5,4))-b(2)*ddP(9,4) , -b(1)*(ddP(8,5)-ddP(5,5
          -b(1)*(ddP(8,4)-ddP(5,4))-b(2)*ddP(9,4) , -b(1)*(ddP(8,2)-ddP(5,2))-b(2)*ddP(9,2) , -b(1)*(ddP(8,6)-ddP(5,6
          -b(1)*(ddP(8,5)-ddP(5,5))-b(2)*ddP(9,5) , -b(1)*(ddP(8,6)-ddP(5,6))-b(2)*ddP(9,6) , -b(1)*(ddP(8,3)-ddP(5,3

ddN33 = [-c(1)*(ddP(8,1)-ddP(5,1))-c(2)*ddP(9,1) , -c(1)*(ddP(8,4)-ddP(5,4))-c(2)*ddP(9,4) , -c(1)*(ddP(8,5)-ddP(5,5
          -c(1)*(ddP(8,4)-ddP(5,4))-c(2)*ddP(9,4) , -c(1)*(ddP(8,2)-ddP(5,2))-c(2)*ddP(9,2) , -c(1)*(ddP(8,6)-ddP(5,6
          -c(1)*(ddP(8,5)-ddP(5,5))-c(2)*ddP(9,5) , -c(1)*(ddP(8,6)-ddP(5,6))-c(2)*ddP(9,6) , -c(1)*(ddP(8,3)-ddP(5,3
```

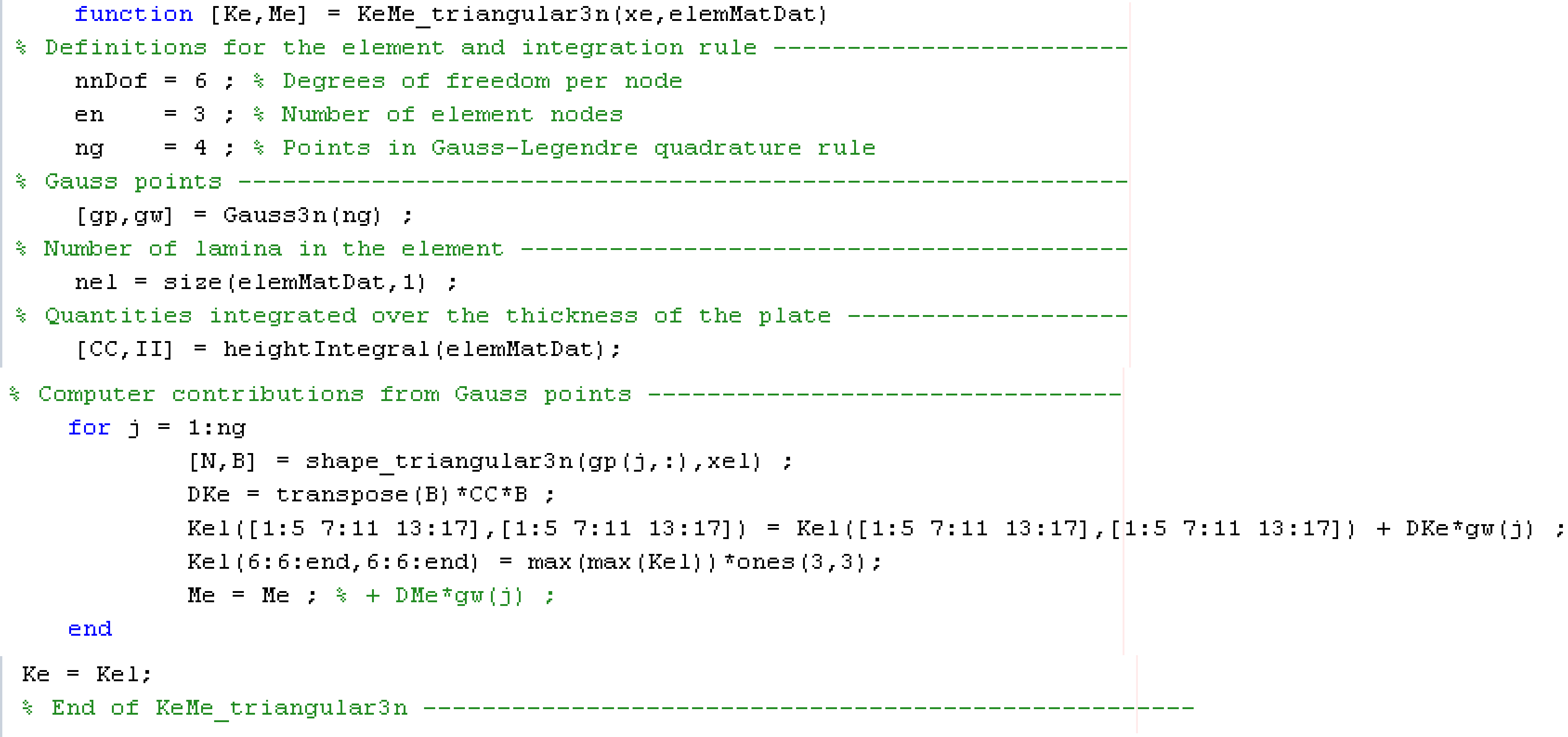
Solution exercise 5, continued

```
A = 1/(2*Delta)*[b(1) b(2) b(3) ; c(1) c(2) c(3)];

ddN11 = A*ddN11*A'; % eq (4.58) p. 132 copy
ddN12 = A*ddN12*A';
ddN13 = A*ddN13*A';
ddN21 = A*ddN21*A';
ddN22 = A*ddN22*A';
ddN23 = A*ddN23*A';
ddN31 = A*ddN31*A';
ddN32 = A*ddN32*A';
ddN33 = A*ddN33*A';
```



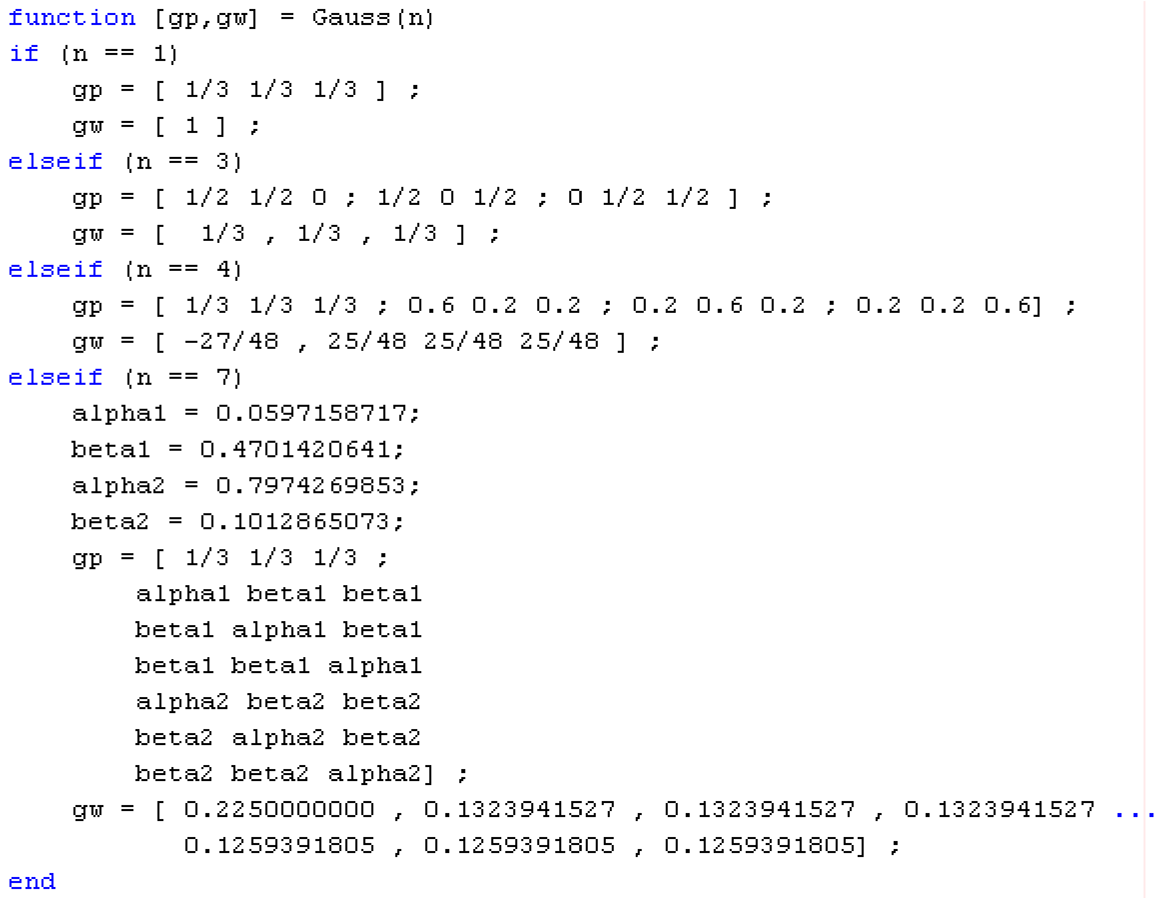
```
B = [0 0 0            0            0            0 0 0            0            0            0 0 0            0            0           ;
     0 0 0            0            0            0 0 0            0            0            0 0 0            0            0           ;
     0 0 0            0            0            0 0 0            0            0            0 0 0            0            0           ;
     0 0 ddN11(1,1)   ddN12(1,1)   ddN13(1,1)   0 0 ddN21(1,1)   ddN22(1,1)   ddN23(1,1)   0 0 ddN31(1,1)   ddN32(1,1)   ddN33(1,1)  ;
     0 0 ddN11(2,2)   ddN12(2,2)   ddN13(2,2)   0 0 ddN21(2,2)   ddN22(2,2)   ddN23(2,2)   0 0 ddN31(2,2)   ddN32(2,2)   ddN33(2,2)  ;
     0 0 2*ddN11(1,2) 2*ddN12(1,2) 2*ddN13(1,2) 0 0 2*ddN21(1,2) 2*ddN22(1,2) 2*ddN23(1,2) 0 0 2*ddN31(1,2) 2*ddN32(1,2) 2*ddN33(1,2)] ;
```

Solution exercise 6.1



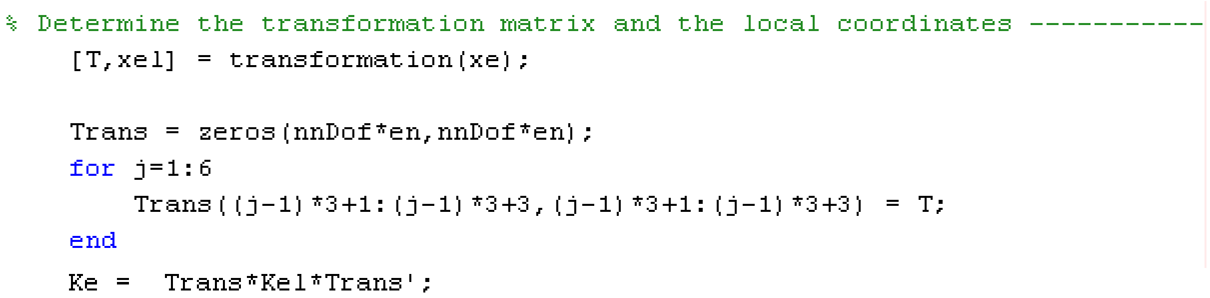
```
    function [Ke,Me] = KeMe_triangular3n(xe,elemMatDat)
% Definitions for the element and integration rule ------------------------
    nnDof = 6 ; % Degrees of freedom per node
    en    = 3 ; % Number of element nodes
    ng    = 4 ; % Points in Gauss-Legendre quadrature rule
% Gauss points ------------------------------------------------------------
    [gp,gw] = Gauss3n(ng) ;
% Number of lamina in the element -----------------------------------------
    nel = size(elemMatDat,1) ;
% Quantities integrated over the thickness of the plate -------------------
    [CC,II] = heightIntegral(elemMatDat);
% Computer contributions from Gauss points --------------------------------
    for j = 1:ng
            [N,B] = shape_triangular3n(gp(j,:),xel) ;
            DKe = transpose(B)*CC*B ;
            Kel([1:5 7:11 13:17],[1:5 7:11 13:17]) = Kel([1:5 7:11 13:17],[1:5 7:11 13:17]) + DKe*gw(j) ;
            Kel(6:6:end,6:6:end) = max(max(Kel))*ones(3,3);
            Me = Me ; % + DMe*gw(j) ;
    end
 Ke = Kel;
 % End of KeMe_triangular3n ------------------------------------------------
```

Solution exercise 6.2

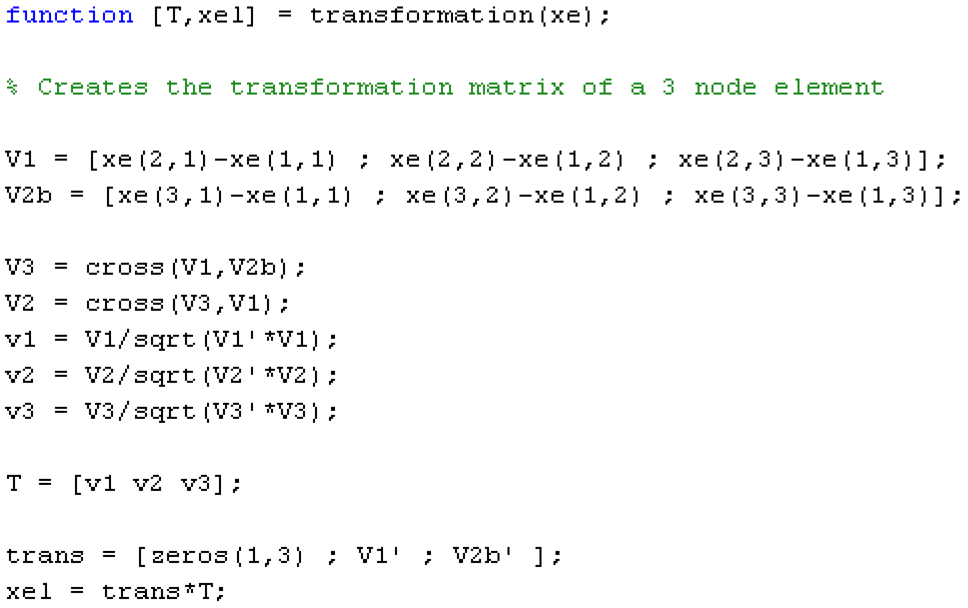


```
function [gp,gw] = Gauss(n)
if (n == 1)
    gp = [ 1/3 1/3 1/3 ] ;
    gw = [ 1 ] ;
elseif (n == 3)
    gp = [ 1/2 1/2 0 ; 1/2 0 1/2 ; 0 1/2 1/2 ] ;
    gw = [  1/3 , 1/3 , 1/3 ] ;
elseif (n == 4)
    gp = [ 1/3 1/3 1/3 ; 0.6 0.2 0.2 ; 0.2 0.6 0.2 ; 0.2 0.2 0.6] ;
    gw = [ -27/48 , 25/48 25/48 25/48 ] ;
elseif (n == 7)
    alpha1 = 0.0597158717;
    beta1 = 0.4701420641;
    alpha2 = 0.7974269853;
    beta2 = 0.1012865073;
    gp = [ 1/3 1/3 1/3 ;
        alpha1 beta1 beta1
        beta1 alpha1 beta1
        beta1 beta1 alpha1
        alpha2 beta2 beta2
        beta2 alpha2 beta2
        beta2 beta2 alpha2] ;
    gw = [ 0.2250000000 , 0.1323941527 , 0.1323941527 , 0.1323941527 ...
           0.1259391805 , 0.1259391805 , 0.1259391805] ;
end
```

Solution exercise 7



```
% Determine the transformation matrix and the local coordinates -----------
    [T,xel] = transformation(xe);

    Trans = zeros(nnDof*en,nnDof*en);
    for j=1:6
        Trans((j-1)*3+1:(j-1)*3+3,(j-1)*3+1:(j-1)*3+3) = T;
    end
    Ke =  Trans*Kel*Trans';
```



```
function [T,xel] = transformation(xe);

% Creates the transformation matrix of a 3 node element

V1 = [xe(2,1)-xe(1,1) ; xe(2,2)-xe(1,2) ; xe(2,3)-xe(1,3)];
V2b = [xe(3,1)-xe(1,1) ; xe(3,2)-xe(1,2) ; xe(3,3)-xe(1,3)];

V3 = cross(V1,V2b);
V2 = cross(V3,V1);
v1 = V1/sqrt(V1'*V1);
v2 = V2/sqrt(V2'*V2);
v3 = V3/sqrt(V3'*V3);

T = [v1 v2 v3];

trans = [zeros(1,3) ; V1' ; V2b' ];
xel = trans*T;
```

Solution exercise 8

```
N1 = L(1);
N2 = L(2);
N3 = L(3);

N = [N1 0   0   0   0   N2 0   0   0   0   N3 0   0   0   0   ;
     0  N1  0   0   0   0  N2  0   0   0   0  N3  0   0   0   ;
     0  0   N11 N12 N13 0  0   N21 N22 N23 0  0   N31 N32 N33 ];
```

```
dN1dx = b(1)/(2*Delta);
dN1dy = c(1)/(2*Delta);
dN2dx = b(2)/(2*Delta);
dN2dy = c(2)/(2*Delta);
dN3dx = b(3)/(2*Delta);
dN3dy = c(3)/(2*Delta);
```

```
B = [dN1dx 0     0           0           0           dN2dx 0     0           0           0           dN3dx 0     0           0           0           ;
     0     dN1dy 0           0           0           0     dN2dy 0           0           0           0     dN3dy 0           0           0           ;
     dN1dy dN1dx 0           0           0           dN2dy dN2dx 0           0           0           dN3dy dN3dx 0           0           0           ;
     0     0     ddN11(1,1)  ddN12(1,1)  ddN13(1,1)  0     0     ddN21(1,1)  ddN22(1,1)  ddN23(1,1)  0     0     ddN31(1,1)  ddN32(1,1)  ddN33(1,1)  ;
     0     0     ddN11(2,2)  ddN12(2,2)  ddN13(2,2)  0     0     ddN21(2,2)  ddN22(2,2)  ddN23(2,2)  0     0     ddN31(2,2)  ddN32(2,2)  ddN33(2,2)  ;
     0     0     2*ddN11(1,2) 2*ddN12(1,2) 2*ddN13(1,2) 0   0     2*ddN21(1,2) 2*ddN22(1,2) 2*ddN23(1,2) 0   0     2*ddN31(1,2) 2*ddN32(1,2) 2*ddN33(1,2)]
```