FINITE ELEMENT METHOD

1 Finite element method for 3D deformation

1.1 Discretisation

The integral over the volume V is written as a sum of integrals over smaller volumes, which collectively constitute the whole volume. Such a small volume V^e is called an element. Subdividing the volume implies that also the surface with area A is subdivided in element surfaces (faces) with area A^e.

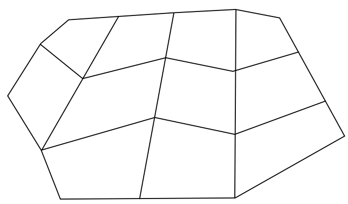

Fig. 1.1 : *Finite element discretisation*

$$\sum_e \int_{V^e} (\vec{\nabla}\vec{w})^T : {}^4\boldsymbol{C} : (\vec{\nabla}\vec{u})\, dV^e = \sum_e \int_{V^e} \vec{w} \cdot \rho\vec{q}\, dV^e + \sum_{e_A} \int_{A^e} \vec{w} \cdot \vec{p}\, dA^e \qquad \forall\ \vec{w}$$

1.2 Isoparametric elements

Each point of a three-dimensional element can be identified with three local coordinates $\{\xi_1, \xi_2, \xi_3\}$. In two dimensions we need two and in one dimension only one local coordinate.

The real geometry of the element can be considered to be the result of a deformation from the original cubic, square or line element with (side) length 2. The deformation can be described with a deformation matrix, which is called the *Jacobian matrix* $\underline{J}$. The determinant of this matrix relates two infinitesimal volumes, areas or lengths of both element representations.

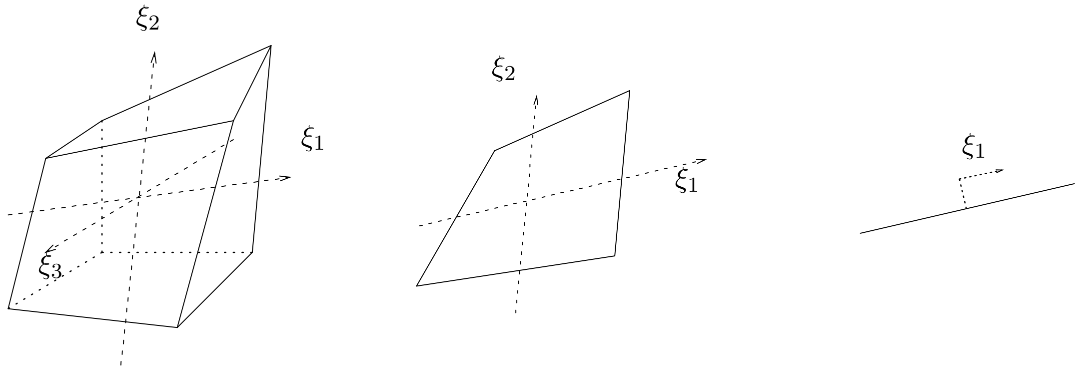

Fig. 1.2 : *Isoparametric elements*

isoparametric (local) coordinates $\quad (\xi_1, \xi_2, \xi_3) \quad ; \quad -1 \leq \xi_i \leq 1 \quad i = 1, 2, 3$

Jacobian matrix $\quad \underline{J} = \left(\nabla_{\underset{\sim}{\xi}} \underset{\sim}{x}^T \right)^T \quad ; \quad dV^e = \det(\underline{J})\, d\xi_1 d\xi_2 d\xi_3$

1.3 Interpolation : 4-node linear element

When the elements have a simple shape, e.g. an six-sided volume, the shape and thus volume is known when the position of a discrete number of edge points is known. These points are the element nodal points. For a cube with plane faces, eight corner points are needed. In two dimensions quadrilaterals with straight edges can be used, where four corner nodes describe the shape. The position of an internal point of the element can be expressed in the position of the *nodal points*. This interpolation is done with so-called shape- or interpolation functions, which are a function of local element coordinates $\xi_i, i = 1, 2, 3$, which assume values between -1 and +1.

The value of the unknown quantity – here the displacement vector $\vec{u}$ – in an arbitrary point of the element, can also be interpolated between the values of that quantity in the element nodes.

Besides $\vec{x}$ and $\vec{u}$, the weighting function $\vec{w}$ also needs to be interpolated between nodal values. When this interpolation is the same as that for the displacement, the so-called *Galerkin* procedure is followed, which is generally the case for simple elements, considered here.

The gradient of $\vec{u}$ and $\vec{w}$ also has to be elaborated. and can be written as the product of a column which contains the derivatives of the interpolation functions, and the column with nodal components of $\vec{u}$ and $\vec{u}$.

Finally, everything is substituted in the weighted residual integral. The volume integral in the left hand side is the element stiffness matrix $\underline{\boldsymbol{K}}^e$. The integrals in the right hand side represent the external load and are summarized in the column $\underset{\sim}{f}^e_e$.

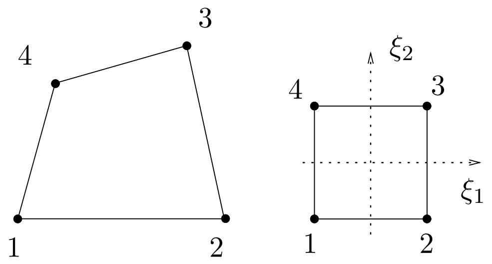

Fig. 1.3 : *Quadrilateral element with four corner nodes*

$$\vec{x} = N^1(\underset{\sim}{\xi})\,\vec{x}^1 + N^2(\underset{\sim}{\xi})\,\vec{x}^2 + N^3(\underset{\sim}{\xi})\,\vec{x}^3 + N^4(\underset{\sim}{\xi})\,\vec{x}^4$$

interpolation functions

$$\begin{aligned} N^1 &= \tfrac{1}{4}(\xi_1 - 1)(\xi_2 - 1) \quad ; \quad & N^2 &= -\tfrac{1}{4}(\xi_2 + 1)(\xi_2 - 1) \\ N^3 &= \tfrac{1}{4}(\xi_1 + 1)(\xi_2 + 1) \quad ; \quad & N^4 &= -\tfrac{1}{4}(\xi_1 - 1)(\xi_2 + 1) \end{aligned}$$

displacement

$$\vec{u} = N^1(\underset{\sim}{\xi})\,\vec{u}^1 + N^2(\underset{\sim}{\xi})\,\vec{u}^2 + N^3(\underset{\sim}{\xi})\,\vec{u}^3 + N^4(\underset{\sim}{\xi})\,\vec{u}^4 = \underset{\sim}{N}^T(\underset{\sim}{\xi})\underset{\sim}{\vec{u}}^e$$

Galerkin

$$\vec{w} = N^1(\underset{\sim}{\xi})\,\vec{w}^1 + N^2(\underset{\sim}{\xi})\,\vec{w}^2 + N^3(\underset{\sim}{\xi})\,\vec{w}^3 + N^4(\underset{\sim}{\xi})\,\vec{w}^4 = \underset{\sim}{N}^T(\underset{\sim}{\xi})\underset{\sim}{\vec{w}}^e$$

gradients

$$\vec{\nabla}\vec{u} = (\vec{\nabla}\underset{\sim}{N})^T\underset{\sim}{\vec{u}}^e = \underset{\sim}{\vec{B}}^T\underset{\sim}{\vec{u}}^e \qquad ; \qquad \vec{\nabla}\vec{w} = (\vec{\nabla}\underset{\sim}{N})^T\underset{\sim}{\vec{w}}^e = \underset{\sim}{\vec{B}}^T\underset{\sim}{\vec{w}}^e$$

weighted residual integral

$$\int_{V^e} (\underset{\sim}{\vec{B}}^T\underset{\sim}{\vec{w}}^e)^T : {}^4\boldsymbol{C} : (\underset{\sim}{\vec{B}}^T\underset{\sim}{\vec{u}}^e)\,dV^e = \int_{V^e} \underset{\sim}{\vec{w}}^{e^T}\underset{\sim}{N}\cdot\rho\vec{q}\,dV^e + \int_{A^e} \underset{\sim}{\vec{w}}^{e^T}\underset{\sim}{N}\cdot\vec{p}\,dA^e$$

$$\underset{\sim}{\vec{w}}^{e^T}\cdot\left[\int_{V^e}\underset{\sim}{\vec{B}}\cdot{}^4\boldsymbol{C}\cdot\underset{\sim}{\vec{B}}^T\,dV^e\right]\cdot\underset{\sim}{\vec{u}}^e = \underset{\sim}{\vec{w}}^{e^T}\cdot\left[\int_{V^e}\underset{\sim}{N}\,\rho\vec{q}\,dV^e\right] + \underset{\sim}{\vec{w}}^{e^T}\cdot\left[\int_{A^e}\underset{\sim}{N}\,\vec{p}\,dA^e\right] \quad \rightarrow$$

$$\underset{\sim}{\vec{w}}^{e^T}\cdot\underline{\boldsymbol{K}}^e\cdot\underset{\sim}{\vec{u}}^e = \underset{\sim}{\vec{w}}^{e^T}\cdot\underset{\sim}{\vec{f}}_e^{\,e}$$

1.4 Integration

Calculating the element stiffness matrix $\underline{K}^e$ and the external loads $\underset{\sim}{f}_e^{\,e}$ implies the evaluation of an integral over the element volume V^e and the element surface A^e. This integration is done numerically, using a fixed set of *nip* Gauss-points, which have s specific location in the element. The value of the integrand is calculated in each Gauss-point and multiplied with a Gauss-point-specific weighting factor c^{ip} and added.

$$\int\limits_{V^e} g(x_1,x_2,x_3)\, dV^e = \int\limits_{\xi_1=-1}^{1}\int\limits_{\xi_2=-1}^{1}\int\limits_{\xi_3=-1}^{1} f(\xi_1,\xi_2,\xi_3)\, d\xi_1 d\xi_2 d\xi_3 = \sum_{ip=1}^{nip} c^{ip}\, f(\xi_1^{ip},\xi_2^{ip},\xi_3^{ip})$$

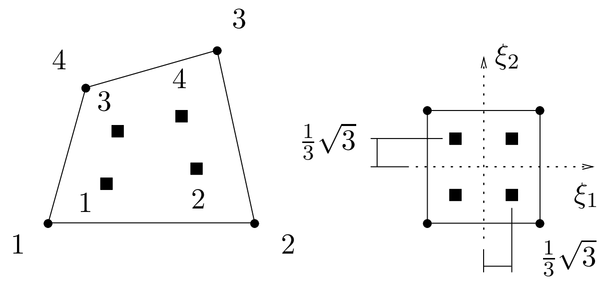

Fig. 1.4 : *Integration points in 4-node quadrilateral element*

$$\int\limits_{V^e} g(x_1,x_2)\, dV^e = \int\limits_{\xi_1=-1}^{1}\int\limits_{\xi_2=-1}^{1} f(\xi_1,\xi_2)\, d\xi_1 d\xi_2 = \sum_{ip=1}^{4} c^{ip}\, f(\xi_1^{ip},\xi_2^{ip})$$

1.5 Assembling

The weighted residual contribution of all elements have to be collected into the total weighted residual integral. This means that all elements are connected or assembled. This assembling is an administrative procedure. All the element matrices and columns are placed at appropriate locations into the structural or global stiffness matrix $\underline{K}$ and the load column $\underset{\sim}{f}_e$.

Because the resulting equation has to be satisfied for all $\underset{\sim}{w}$, the nodal displacements $\underset{\sim}{u}$ have to satisfy a set of equations.

$$\sum_e \vec{\underset{\sim}{w}}^{e^T} \cdot \underline{\boldsymbol{K}}^e \cdot \vec{\underset{\sim}{u}}^e = \sum_e \vec{\underset{\sim}{w}}^{e^T} \cdot \vec{\underset{\sim}{f}}_e^e \quad \rightarrow \quad \vec{\underset{\sim}{w}}^T \cdot \underline{\boldsymbol{K}} \cdot \vec{\underset{\sim}{u}} = \vec{\underset{\sim}{w}}^T \cdot \vec{\underset{\sim}{f}}_e \quad \forall\, \vec{\underset{\sim}{w}} \qquad \rightarrow$$

$$\underline{\boldsymbol{K}} \cdot \vec{\underset{\sim}{u}} = \vec{\underset{\sim}{f}}_e \quad \rightarrow \quad \vec{\underset{\sim}{u}} = \underline{\boldsymbol{K}}^{-1} \cdot \vec{\underset{\sim}{f}}_e$$

1.6 Boundary conditions

The initial governing equations were differential equations, which obviously need boundary conditions to arrive at a unique solution. The boundary conditions are prescribed displacements or forces in certain material points. After finite element discretisation, displacements and forces can be applied in nodal points.

The set of nodal equations $\underline{K}\underset{\sim}{u} = \underset{\sim}{f}_e$ cannot be solved yet, because the structural stiffness matrix $\underline{K}$ is singular and cannot be inverted. First some essential boundary conditions must be applied, which prevent the rigid body motion of the material and renders the equations solvable.

2 Program structure

A finite element program starts with reading data from an input file and initialization of variables and databases.

First the system of equations is build. In a loop over all elements, the stresses are calculated and the material stiffness is updated. The element internal nodal force column and the element stiffness matrix are assembled into the global column and matrix.

Numerical integration over the element volume (area) implies that a loop over integration points is entered. In each integration point the integrand is calculated and the result is multiplied by a weighting factor and added to the existing element value.

After taking tyings and boundary conditions into account, the unknown nodal displacements and reaction forces are calculated.

```
read input data from input file
calculate additional variables from input data
initialize values and arrays


   for all elements
      for all integration points
         calculate contribution to initial element stiffness matrix
      end integration point loop
      assemble global stiffness matrix
   end element loop

   determine external incremental load from input


      take tyings into account
      take boundary conditions into account

      calculate iterative nodal displacements
      calculate total deformation

      for all elements
         for all integration points
            calculate stresses from material behavior
            calculate material stiffness from material behavior
            calculate contribution to element internal nodal forces
            calculate contribution to element stiffness matrix
         end ntegration point loop
         assemble global stiffness matrix
         assemble global internal load column
      end element loop
```

```
    calculate residual load column
    calculate convergence norm


 store data for post-processing
```

A more detailed description of the formulation of axi-symmetric ring elements and planar elements can be found in the appendices **??** **??**, **??**, 3. In the next chapter, some results are presented for linear elastic problems.

3 Planar elements

The 4-noded and 8-noded elements are described in the next sections of this appendix.

3.1 Four-node quadrilateral element

In two-dimensional finite element analysis the four-node element is used very much. In the undeformed and deformed configuration the element sides are straight lines. As its name indicates, it has four nodal points, which are located in its corners. The numbering of the nodes is anti-clockwise by convention.

The shape functions, which are used to interpolate global coordinates and displacement components and weighting function components between their respective nodal values, must be linear along an element side. In the two-dimensional plane these functions are functions of the isoparametric coordinates ξ_1 and ξ_2. These functions are not completely linear : they have a term $\xi_1\xi_2$. This implies that their derivatives are not completely constant.

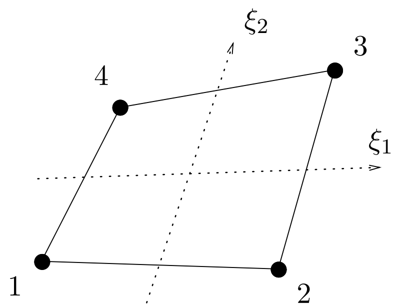

Fig. 3.5 : *Four-node quadrilateral element*

$$
\begin{array}{lcl}
N^1 = \frac{1}{4}(1-\xi_1)(1-\xi_2) & ; & N^2 = \frac{1}{4}(1+\xi_1)(1-\xi_2) \\
N^3 = \frac{1}{4}(1+\xi_1)(1+\xi_2) & ; & N^4 = \frac{1}{4}(1-\xi_1)(1+\xi_2)
\end{array}
$$

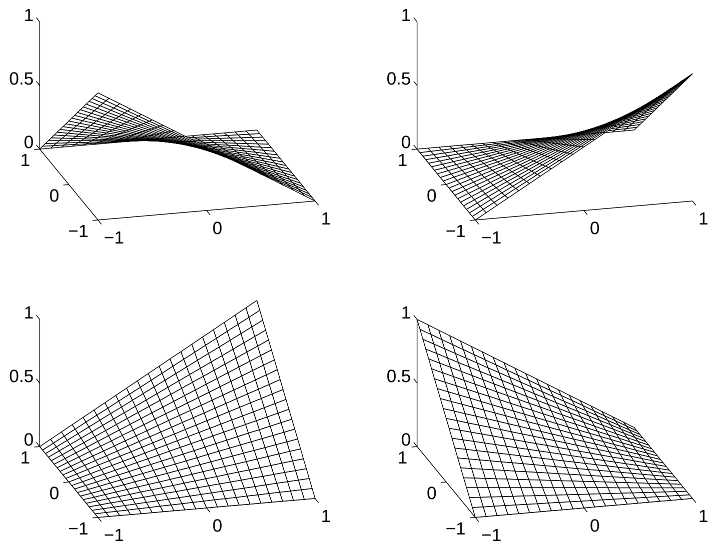

Fig. 3.6 : *Linear interpolation functions in 4-node element*

3.2 Cartesian coordinate system

In a Cartesian coordinate system the displacement of every point of a quadrilateral element has two components, u_x and u_y. Both components are interpolated between the nodal displacement components, using the shape functions. The element shape and the weighting function is interpolated in the same way as the displacement.

displacement

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} N^1 & 0 & N^2 & 0 & N^3 & 0 & N^4 & 0 \\ 0 & N^1 & 0 & N^2 & 0 & N^3 & 0 & N^4 \end{bmatrix} \begin{bmatrix} u_x^1 \\ u_y^1 \\ u_x^2 \\ u_y^2 \\ u_x^3 \\ u_y^3 \\ u_x^4 \\ u_y^4 \end{bmatrix} \rightarrow \underset{\sim}{u} = \underline{N}\,\underset{\sim}{u}^e$$

element shape $\quad \underset{\sim}{x} = \underline{N}\,\underset{\sim}{x}^e \quad ; \quad \underset{\sim}{x}_0 = \underline{N}\,\underset{\sim}{x}_0^e$

weighting function $\quad \underset{\sim}{w} = \underline{N}\,\underset{\sim}{w}^e$

The columns $\underset{\approx}{L}_u$ and $\underset{\approx}{L}_w$ represent the components of the gradient of the displacement and the weighting function, respectively. After interpolation of displacement and weighting function, the so-called B-matrix appears.

The B-matrix contains derivatives of the shape functions $\{N^\alpha; \alpha = 1,2,3,4\}$ with respect to the Cartesian coordinates x and y. The Jacobian matrix $\underline{J}$ contains the derivatives of the Cartesian coordinates x and y with respect to the isoparametric coordinates ξ_1 and ξ_2.

$$
\begin{bmatrix} u_{x,x} \\ u_{y,y} \\ u_{y,x} \\ u_{x,y} \end{bmatrix} = \begin{bmatrix} N^1_{,x} & 0 & N^2_{,x} & 0 & N^3_{,x} & 0 & N^4_{,x} & 0 \\ 0 & N^1_{,y} & 0 & N^2_{,y} & 0 & N^3_{,y} & 0 & N^4_{,y} \\ 0 & N^1_{,x} & 0 & N^2_{,x} & 0 & N^3_{,x} & 0 & N^4_{,x} \\ N^1_{,y} & 0 & N^2_{,y} & 0 & N^3_{,y} & 0 & N^4_{,y} & 0 \end{bmatrix} \begin{bmatrix} u^1_x \\ u^1_y \\ u^2_x \\ u^2_y \\ u^3_x \\ u^3_y \\ u^4_x \\ u^4_y \end{bmatrix} \rightarrow \left(\underset{\approx}{L}_u \right)_t = \underline{B}\, \underset{\sim}{u}^e
$$

$$
\begin{bmatrix} N^1_{,x} & N^1_{,y} \\ N^2_{,x} & N^2_{,y} \\ N^3_{,x} & N^3_{,y} \\ N^4_{,x} & N^4_{,y} \end{bmatrix} = \begin{bmatrix} N^1_{,1} & N^1_{,2} \\ N^2_{,1} & N^2_{,2} \\ N^3_{,1} & N^3_{,2} \\ N^4_{,1} & N^4_{,2} \end{bmatrix} \begin{bmatrix} \xi_{1,x} & \xi_{1,y} \\ \xi_{2,x} & \xi_{2,y} \end{bmatrix} = \begin{bmatrix} N^1_{,1} & N^1_{,2} \\ N^2_{,1} & N^2_{,2} \\ N^3_{,1} & N^3_{,2} \\ N^4_{,1} & N^4_{,2} \end{bmatrix} \underline{J}^{-T}
$$

$$
\underline{J} = \begin{bmatrix} x_{,1} & y_{,1} \\ x_{,2} & y_{,2} \end{bmatrix} = \begin{bmatrix} N^1_{,1} & N^2_{,1} & N^3_{,1} & N^4_{,1} \\ N^1_{,2} & N^2_{,2} & N^3_{,2} & N^4_{,2} \end{bmatrix} \begin{bmatrix} x^1_e & y^1_e \\ x^2_e & y^2_e \\ x^3_e & y^3_e \\ x^4_e & y^4_e \end{bmatrix}
$$

The deformation matrix can be calculated in each element integration point. Besides nodal point coordinates in the current state, the coordinates in the reference state must be available.

$$
\underline{F} = \begin{bmatrix} \dfrac{\partial x}{\partial x_0} & \dfrac{\partial x}{\partial y_0} & 0 \\ \dfrac{\partial y}{\partial x_0} & \dfrac{\partial y}{\partial y_0} & 0 \\ 0 & 0 & F_{zz} \end{bmatrix}
$$

$$
\begin{aligned}
\begin{bmatrix} \dfrac{\partial x}{\partial x_0} & \dfrac{\partial y}{\partial x_0} \\ \dfrac{\partial x}{\partial y_0} & \dfrac{\partial y}{\partial y_0} \end{bmatrix} &= \begin{bmatrix} N^1_{,x0} & N^2_{,x0} & N^3_{,x0} & N^4_{,x0} \\ N^1_{,y0} & N^2_{,y0} & N^3_{,y0} & N^4_{,y0} \end{bmatrix} \begin{bmatrix} x^1_e & y^1_e \\ x^2_e & y^2_e \\ x^3_e & y^3_e \\ x^4_e & y^4_e \end{bmatrix} \\
&= \begin{bmatrix} \xi_{1,x0} & \xi_{2,x0} \\ \xi_{1,y0} & \xi_{2,y0} \end{bmatrix} \begin{bmatrix} N^1_{,1} & N^2_{,1} & N^3_{,1} & N^4_{,1} \\ N^1_{,2} & N^2_{,2} & N^3_{,2} & N^4_{,2} \end{bmatrix} \begin{bmatrix} x^1_e & y^1_e \\ x^2_e & y^2_e \\ x^3_e & y^3_e \\ x^4_e & y^4_e \end{bmatrix} = \underline{J}_0^{-1} \underline{J}
\end{aligned}
$$

3.3 Cylindrical coordinate system

In a cylindrical coordinate system the displacement of every point of a quadrilateral element has two components, u_r and u_z. Both components are interpolated between the nodal displacement components, using the shape functions. The element shape and the weighting function is interpolated in the same way as the displacement.

displacement

$$\begin{bmatrix} u_r \\ u_z \end{bmatrix} = \begin{bmatrix} N^1 & 0 & N^2 & 0 & N^3 & 0 & N^4 & 0 \\ 0 & N^1 & 0 & N^2 & 0 & N^3 & 0 & N^4 \end{bmatrix} \begin{bmatrix} u_r^1 \\ u_z^1 \\ u_r^2 \\ u_z^2 \\ u_r^3 \\ u_z^3 \\ u_r^4 \\ u_z^4 \end{bmatrix} \rightarrow \underset{\sim}{u} = \underline{N}\,\underset{\sim}{u}_e$$

element shape $\qquad r = \underset{\sim}{N}^T \underset{\sim}{r} \quad ; \quad z = \underset{\sim}{N}^T \underset{\sim}{z}$

$\qquad r_0 = \underset{\sim}{N}^T \underset{\sim}{r}_0 \quad ; \quad z_0 = \underset{\sim}{N}^T \underset{\sim}{z}_0$

weighting function $\qquad \underset{\sim}{w} = \underline{N}\,\underset{\sim}{w}^e$

The columns $\underset{\approx u}{L}$ and $\underset{\approx w}{L}$ represent the components of the gradient of the displacement and the weighting function, respectively. After interpolation of displacement and weighting function, the so-called B-matrix appears. For ease of programming, we swap the derivatives $u_{t,t}$ and $u_{z,z}$ in the column $\underset{\approx u}{L}$ and analoguously $w_{t,t}$ and $w_{z,z}$ in the column $\underset{\approx w}{L}$.

The B-matrix contains derivatives of the shape functions $\{N^\alpha; \alpha = 1,2,3,4\}$ with respect to the cylindrical coordinates r and z. The Jacobian matrix $\underline{J}$ contains the derivatives of the cylindrical coordinates r and z with respect to the isoparametric coordinates ξ_1 and ξ_2.

$$\begin{bmatrix} u_{r,r} \\ u_{z,z} \\ \frac{1}{r}u_r \\ u_{z,r} \\ u_{r,z} \end{bmatrix} = \begin{bmatrix} N^1_{,r} & 0 & N^2_{,r} & 0 & N^3_{,r} & 0 & N^4_{,r} & 0 \\ 0 & N^1_{,z} & 0 & N^2_{,z} & 0 & N^3_{,z} & 0 & N^4_{,z} \\ \frac{1}{r}N^1 & 0 & \frac{1}{r}N^2 & 0 & \frac{1}{r}N^3 & 0 & \frac{1}{r}N^4 & 0 \\ 0 & N^1_{,r} & 0 & N^2_{,r} & 0 & N^3_{,r} & 0 & N^4_{,r} \\ N^1_{,z} & 0 & N^2_{,z} & 0 & N^3_{,z} & 0 & N^4_{,z} & 0 \end{bmatrix} \begin{bmatrix} u_r^1 \\ u_z^1 \\ u_r^2 \\ u_z^2 \\ u_r^3 \\ u_z^3 \\ u_r^4 \\ u_z^4 \end{bmatrix} \rightarrow \left(\underset{\approx u}{L}\right)_t = \underline{B}\underset{\sim}{u}^e$$

$$\begin{bmatrix} N^1_{,r} & N^1_{,z} \\ N^2_{,r} & N^2_{,z} \\ N^3_{,r} & N^3_{,z} \\ N^4_{,r} & N^4_{,z} \end{bmatrix} = \begin{bmatrix} N^1_{,1} & N^1_{,2} \\ N^2_{,1} & N^2_{,2} \\ N^3_{,1} & N^3_{,2} \\ N^4_{,1} & N^4_{,2} \end{bmatrix} \begin{bmatrix} \xi_{1,r} & \xi_{1,z} \\ \xi_{2,r} & \xi_{2,z} \end{bmatrix} = \begin{bmatrix} N^1_{,1} & N^1_{,2} \\ N^2_{,1} & N^2_{,2} \\ N^3_{,1} & N^3_{,2} \\ N^4_{,1} & N^4_{,2} \end{bmatrix} \underline{J}^{-T}$$

$$\underline{J} = \begin{bmatrix} r_{,1} & z_{,1} \\ r_{,2} & z_{,2} \end{bmatrix} = \begin{bmatrix} N^1_{,1} & N^2_{,1} & N^3_{,1} & N^4_{,1} \\ N^1_{,2} & N^2_{,2} & N^3_{,2} & N^4_{,2} \end{bmatrix} \begin{bmatrix} r^1_e & z^1_e \\ r^2_e & z^2_e \\ r^3_e & z^3_e \\ r^4_e & z^4_e \end{bmatrix}$$

The deformation matrix can be calculated in each element integration point. Besides nodal point coordinates in the current state, the coordinates in the reference state must be available.

$$\underline{F} = \begin{bmatrix} \dfrac{\partial r}{\partial r_0} & \dfrac{\partial r}{\partial z_0} & 0 \\ \dfrac{\partial z}{\partial r_0} & \dfrac{\partial z}{\partial z_0} & 0 \\ 0 & 0 & \frac{r}{r_0} \end{bmatrix}$$

$$\begin{bmatrix} \dfrac{\partial r}{\partial r_0} & \dfrac{\partial z}{\partial r_0} \\ \dfrac{\partial r}{\partial z_0} & \dfrac{\partial z}{\partial z_0} \end{bmatrix} = \begin{bmatrix} N^1_{,r0} & N^2_{,r0} & N^3_{,r0} & N^4_{,r0} \\ N^1_{,z0} & N^2_{,z0} & N^3_{,z0} & N^4_{,z0} \end{bmatrix} \begin{bmatrix} r^1_e & z^1_e \\ r^2_e & z^2_e \\ r^3_e & z^3_e \\ r^4_e & z^4_e \end{bmatrix}$$

$$= \begin{bmatrix} \xi_{1,r0} & \xi_{2,r0} \\ \xi_{1,z0} & \xi_{2,z0} \end{bmatrix} \begin{bmatrix} N^1_{,1} & N^2_{,1} & N^3_{,1} & N^4_{,1} \\ N^1_{,2} & N^2_{,2} & N^3_{,2} & N^4_{,2} \end{bmatrix} \begin{bmatrix} r^1_e & z^1_e \\ r^2_e & z^2_e \\ r^3_e & z^3_e \\ r^4_e & z^4_e \end{bmatrix} = \underline{J}_0^{-1} \underline{J}$$

3.4 Numerical integration

To generate the element stiffness matrix and the residual force column, integration over the element volume (2D : area) must be carried out. With quadrilateral elements this integration cannot be done analytically, so numerical integration is necessary.

The numerical integration which we employ here is the Gauss quadrature integration. The integrand is evaluated in a number of discrete points, the integration points or Gauss points. The integration point values are multiplied by a weighting factor, ζ, after which they are added. The location of the integration points (= their isoparametric coordinates) and the value of the weighting factor are determined in such a way that a polynomial of a certain degree is integrated exactly.

The four-node quadrilateral integration point locations and weighting factor values are shown in the table. Their choice is such that a polynomial of third order in each direction is integrated exactly.

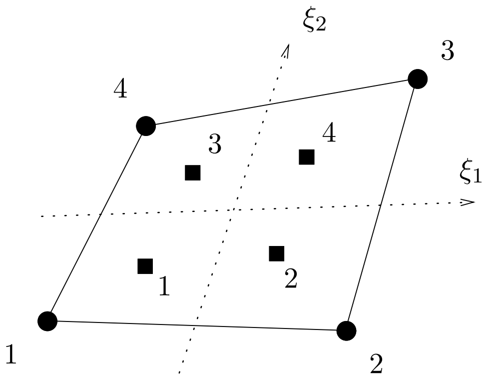

Fig. 3.7 : *Integration points in a 4-node element*

ip	ξ_1	ξ_2	ζ
1	$-\frac{1}{3}\sqrt{3}$	$-\frac{1}{3}\sqrt{3}$	1
2	$\frac{1}{3}\sqrt{3}$	$-\frac{1}{3}\sqrt{3}$	1
3	$-\frac{1}{3}\sqrt{3}$	$\frac{1}{3}\sqrt{3}$	1
4	$\frac{1}{3}\sqrt{3}$	$\frac{1}{3}\sqrt{3}$	1

3.5 Eight-node quadrilateral element

The eight-node element has four sides ("quadrilateral"), which are straight lines in the undeformed configuration. The nodes 1 to 4 are located in the corners (corner nodes), the nodes 5 to 8 are located in the middle of the sides (midpoint nodes). The numbering is anti-clockwise.

Global coordinates, displacement components and weighting function components are interpolated with shape functions which are quadratic along an element side. This implies that in the deformed configuration these sides may be parabolic. The shape functions are functions of the isoparametric coordinates ξ_1 and ξ_2.

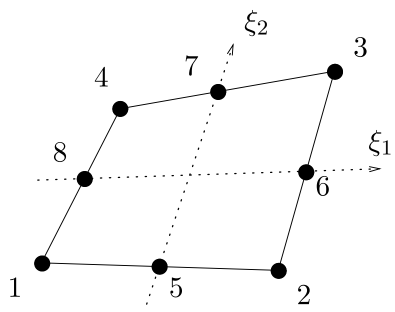

Fig. 3.8 : *Eight-node quadrilateral element*

$$
\begin{aligned}
N^1 &= \tfrac{1}{4}(\xi_1 - 1)(\xi_2 - 1)(-\xi_1 - \xi_2 - 1) & N^5 &= \tfrac{1}{2}(\xi_1^2 - 1)(\xi_2 - 1) \\
N^2 &= \tfrac{1}{4}(\xi_1 + 1)(\xi_2 - 1)(-\xi_1 + \xi_2 + 1) & N^6 &= \tfrac{1}{2}(-\xi_1 - 1)(\xi_2^2 - 1) \\
N^3 &= \tfrac{1}{4}(\xi_1 + 1)(\xi_2 + 1)(\xi_1 + \xi_2 - 1) & N^7 &= \tfrac{1}{2}(\xi_1^2 - 1)(-\xi_2 - 1) \\
N^4 &= \tfrac{1}{4}(\xi_1 - 1)(\xi_2 + 1)(\xi_1 - \xi_2 + 1) & N^8 &= \tfrac{1}{2}(\xi_1 - 1)(\xi_2^2 - 1)
\end{aligned}
$$

The figures show the shape functions associated with the nodal points. The first four plots show the shape functions of the corner nodes and second series of four plots shows those of the mid-side nodes.

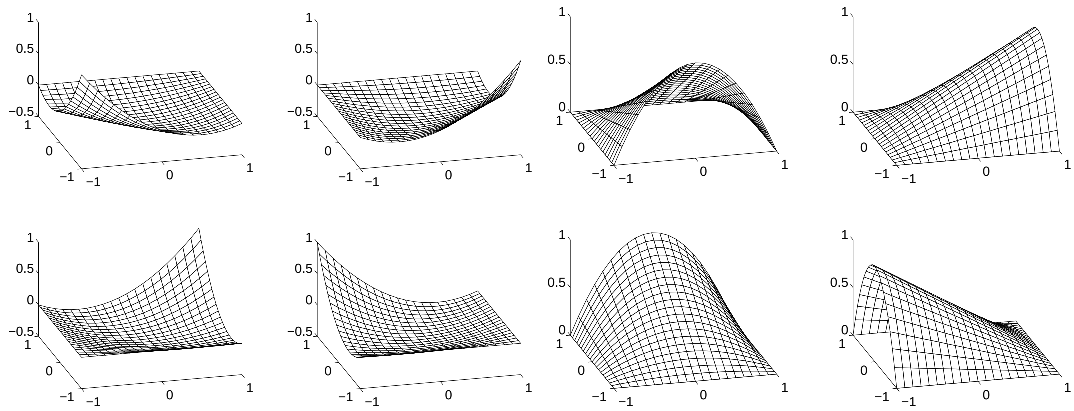

Fig. 3.9 : *Quadratic interpolation functions in 8-node element*

3.6 Numerical integration

The table contains the location of the 9 integration (Gauss) points and their weighting functions for the eight-node quadrilateral element. Their choice is such that a polynomial of fifth order in each direction is integrated exactly.

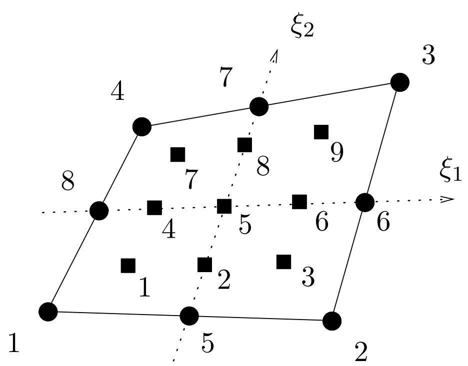

Fig. 3.10 : *Integration points in 8-node element*

$$a = 0.77459 \; ; \quad p = 0.55556 \; ; \quad q = 0.88889$$

ip	ξ_1	ξ_2	ζ
1	$-a$	$-a$	$p \times p$
2	0	$-a$	$p \times q$
3	a	$-a$	$p \times p$
4	$-a$	0	$p \times q$
5	0	0	$q \times q$
6	a	0	$p \times q$
7	$-a$	a	$p \times p$
8	0	a	$p \times q$
9	a	a	$p \times p$

4 FE program `plaxL`

The Matlab program `plaxL` is used to model and analyze linear, planar and axi-symmetric problems. So deformations are small and the material behavior is linear elastic.

4.1 Thick-walled pressurized cylinder: plane stress

A thick-walled open cylinder – inner radius a, outer radius b – is subjected to an internal pressure. Parameter values are listed in the table below. The figures show the element model, both undeformed and deformed., and the plot of the radial displacement over the radius.

First a quarter of the cylinder is modeled and analyzed in plane stress.

$a = 0.25$ m	$b = 0.5$ m	$h = 0.5$ m	$E = 250$ GPa	$\nu = 0.33$
$p_i = 100$ MPa	$p_e = 0$ MPa			

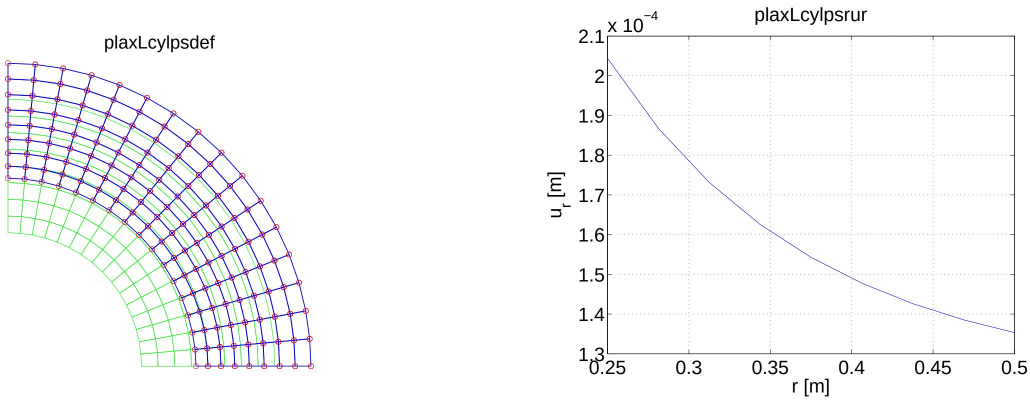

Fig. 4.11 : *Deformation (×1000) of the cylinder and radial displacement.*

The thick-walled open cylinder is now analyzed as an axi-symmetric model.

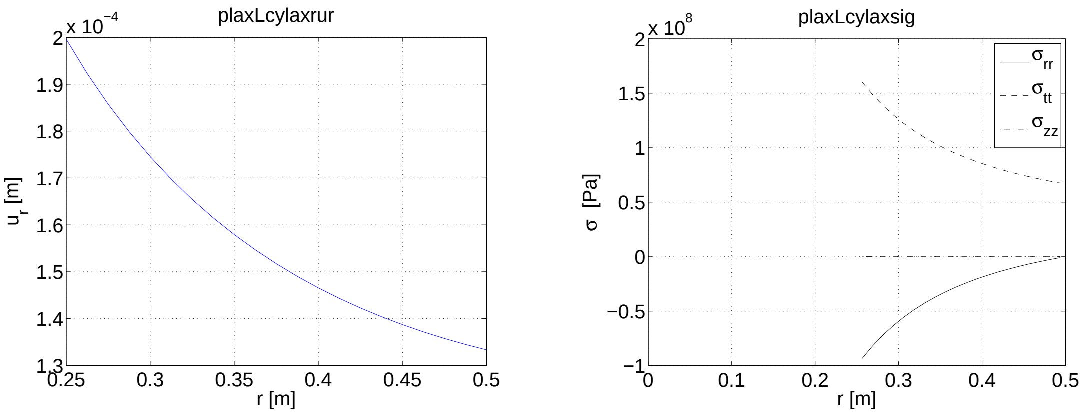

Fig. 4.12 : *Displacement and stresses in a pressurized cylinder.*