FINITE ELEMENT METHOD

1 Finite element method

In Chapter **??** the weighted residual integral was derived.

$$\int_V \left(\underset{\approx}{L}_w\right)_t^T [\underline{\underline{W}}] \left(\underset{\approx}{L}_u\right)_t \, dV = f_e(\underset{\sim}{w}) - f_i(\underset{\sim}{w})$$

The unknown (iterative) dislacements in $\underset{\approx}{L}_u$ have to be determined such that this integral is satisfied for all weighting functions in $\underset{\approx}{L}_w$. Such a solution canot generally be determined exactly and by analytical means. Instead we have to resort to numerical techniques to determine an approximate solution. The finite element method is widely used for this task.

1.1 Discretisation

The integral over the body 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.

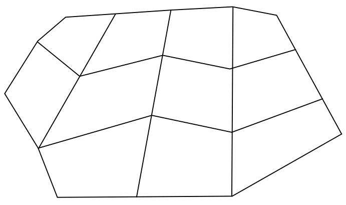

Fig. 1.1 : *Finite element discretisation*

$$\sum_e \int_{V^e} \left(\underset{\approx}{L}_w\right)_t^T [\underline{\underline{W}}] \left(\underset{\approx}{L}_u\right)_t \, dV^e = \sum_e f_e^e(\underset{\sim}{w}) - \sum_e f_i^e(\underset{\sim}{w}) \qquad \forall \; \underset{\sim}{w}$$

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.

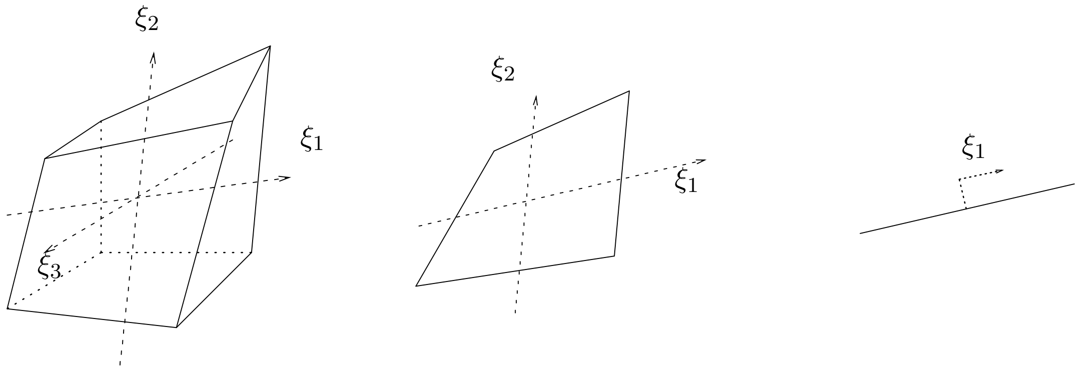

Fig. 1.2 : *Isoparametric elements*

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

1.2 Interpolation

The value of the unknown quantity – here the displacement vector $\vec{u}$ or the iterative displacement vector $\delta\vec{u}$ – in an arbitrary point of the element, can be interpolated between the values of that quantity in certain fixed points of the element : the *element nodes*. *Interpolation functions* N are a function of the isoparametric coordinates.

The components of the vector $(\delta)\vec{u}$ are stored in a column $(\delta)\underset{\sim}{u}$. The nodal (iterative) displacement components are stored in the column $(\delta)\underset{\sim}{u}^e$. The position $\vec{x}$ of a point within the element is interpolated between the nodal point positions, the components of which are stored in the column $\underset{\sim}{x}^e$. Generally, the interpolations for position and displacement are chosen to be the same.

Besides $(\delta)\vec{u}$ and $\vec{x}$, 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.

We consider the vector function $\vec{a}$ to be interpolated, where *nep* is the number of element nodes. Components a_i of $\vec{a}$ w.r.t. a global vector base, can then also be interpolated.

$$\vec{a} = N^1 \vec{a}^1 + N^2 \vec{a}^2 + \cdots + N^{nep} \vec{a}^{nep} = \underset{\sim}{N}^T \underset{\sim}{\vec{a}}^e \quad \rightarrow$$
$$a_i = N^1 a_i^1 + N^2 a_i^2 + \cdots + N^{nep} a_i^{nep} = \sum_{\alpha=1}^{nep} N^\alpha a_i^\alpha = \underset{\sim}{N}^T \underset{\sim}{a}_i^e \quad \rightarrow \quad \underset{\sim}{a} = \underline{N}\, \underset{\sim}{a}^e$$

The gradient of the vector function $\vec{a}$ also has to be elaborated. The gradient is referred to as the second-order tensor $\boldsymbol{L}^c$, which can be written in components w.r.t. a vector basis. The components are stored in a column $\underset{\approx}{L}$. This column can be written as the product of the so-called *B-matrix*, which contains the derivatives of the interpolation functions, and the column with nodal components of $\vec{a}$.

$$\boldsymbol{L}^c = \vec{\nabla}\vec{a} \quad \rightarrow \quad \underset{\approx}{L}_t = \underline{B}\,\underset{\sim}{a}^e$$

1.3 Integration

Interpolations for both the (iterative) displacement and the weighting function and their respective derivatives are substituted in the weighted residual integrals of each element.

Calculating the element contributions 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.

Resulting element quantities are the element stiffness matrix $\underline{K}^e$, the element external force column $\underset{\sim}{f}^e_e$ and the element internal force column $\underset{\sim}{f}^e_i$.

$$\underset{\sim}{w}^{eT}\left[\int\limits_{V^e} \underline{B}^T\,\underline{\underline{W}}\,\underline{B}\,dV^e\right]\delta\underset{\sim}{u}^e = \underset{\sim}{w}^{eT}\underset{\sim}{f}^e_e - \underset{\sim}{w}^{eT}\underset{\sim}{f}^e_i$$

$$\underset{\sim}{w}^{eT}\left[\int\limits_{\xi_1=-1}^{\xi_1=1}\int\limits_{\xi_2=-1}^{\xi_2=1}\int\limits_{\xi_3=-1}^{\xi_3=1} \underline{B}^T\left[\underline{\underline{W}}\right]\underline{B}\;\det(\underline{J})\,d\xi_1 d\xi_2 d\xi_3\right]\delta\underset{\sim}{u}^e = \underset{\sim}{w}^{eT}\underset{\sim}{f}^e_e - \underset{\sim}{w}^{eT}\underset{\sim}{f}^e_i$$

$$\underset{\sim}{w}^{eT}\,\underline{K}^e\,\delta\underset{\sim}{u}^e = \underset{\sim}{w}^{eT}\,\underset{\sim}{f}^e_e - \underset{\sim}{w}^{eT}\,\underset{\sim}{f}^e_i$$

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

1.4 Assemblation

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.

$$\begin{aligned}
&\sum_e \underset{\sim}{w}^{eT}\underline{K}^e\,\delta\underset{\sim}{u}^e = \sum_e \underset{\sim}{w}^{eT}\underset{\sim}{f}^e_e - \sum_e \underset{\sim}{w}^{eT}\underset{\sim}{f}^e_i \quad \rightarrow \\
&\underset{\sim}{w}^T\,\underline{K}\,\delta\underset{\sim}{u} = \underset{\sim}{w}^T\,\underset{\sim}{f}_e - \underset{\sim}{w}^T\,\underset{\sim}{f}_i = \underset{\sim}{w}^T\,\underset{\sim}{r} \qquad \forall\,\underset{\sim}{w} \qquad \rightarrow \\
&\underline{K}\,\delta\underset{\sim}{u} = \underset{\sim}{f}_e - \underset{\sim}{f}_i = \underset{\sim}{r}
\end{aligned}$$

1.5 Solution

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 $\underline{K}\delta \underset{\sim}{u} = \underset{\sim}{r}$ cannot be solved yet, because the structural stiffness matrix $\underline{K}$ is singular and cannot be inverted.

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} & \cdots \\ k_{21} & k_{22} & k_{23} & \cdots \\ k_{31} & k_{32} & k_{33} & \cdots \\ \vdots & \vdots & \vdots & . \end{bmatrix} \begin{bmatrix} a \\ a \\ a \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \quad \rightarrow \quad \underline{K} = \text{ singular} \quad \rightarrow \quad \det \underline{K} = 0$$

When enough boundary conditions have been applied to prevent the rigid body motion of the material body, the equations are solvable.

$$\delta \underset{\sim}{u} = \underline{K}^{-1} \underset{\sim}{r}$$

1.6 Program structure

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

The loading is prescribed as a function of the (fictitious) time in an incremental loop. In each increment the system of nonlinear equilibrium equations is solved iteratively.

In each iteration loop 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.

When the convergence criterion is not reached, a new iteration step is performed. After convergence output data are stored and the next incremental step is carried out.

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

while load increments to be done

   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

   while non-converged iteration step

      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

   end iteration step

   store data for post-processing
end load increment
```

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

1.7 FE program `plax`

The Matlab program **`plax`** is used to model and analyze planar and axisymmetrc problems. Large deformations and rotations are allowed. Various nonlinear and time-dependent material models are implemented .

Rigid rotation

When a material body is subjected to a rigid rotation, no stresses must be generated in the material. The material model must be such that this requirement is satisfied.

In this example we rotate one element over 720^o by prescribing all nodal point displacements. A non-suitable material law, a linear relation between the Cauchy stress tensor $\boldsymbol{\sigma}$ and the infinitesimal strain tensor $\boldsymbol{\varepsilon}$, will result in high reaction forces in the nodes (figure 1 below). A correctly formulated material law, such as a linear relation between the Cauvhy stress tensor $\boldsymbol{\sigma}$ and the tensor $\frac{1}{2}(\boldsymbol{F} \cdot \boldsymbol{F}^c - \boldsymbol{I})$, will result in zero reaction forces (figure 2 below).

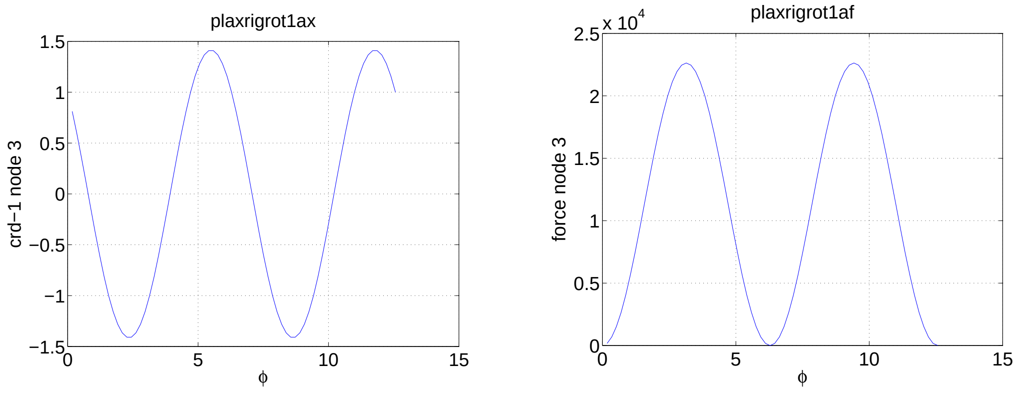

Fig. 1.3 : *Rigid rotation for non-objective elastic material model.*

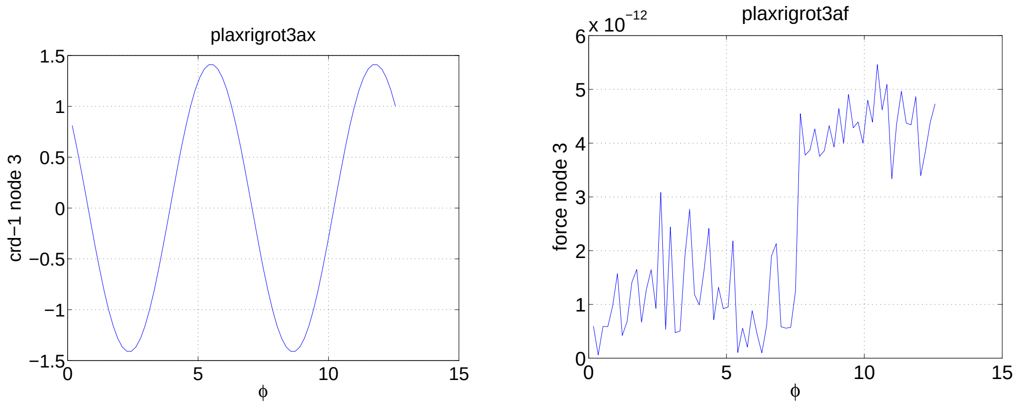

Fig. 1.4 : *Rigid rotation for objective elastic material model.*