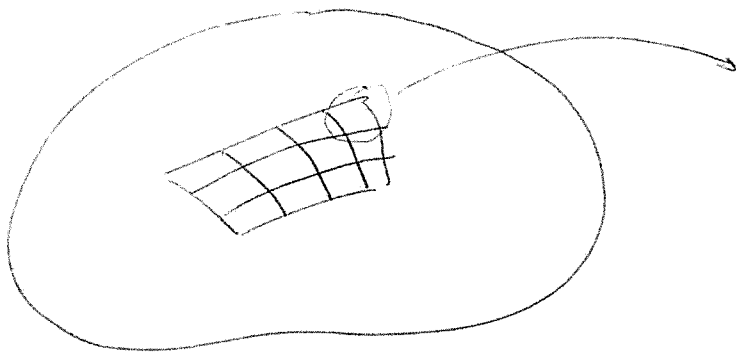
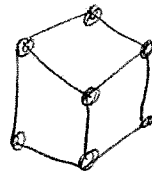


Introduction to the Finite Element Method

The main idea of the finite element consists in dividing the continuum - a three dimensional one as well as a plate/shell or beam can be considered - into an assembly of small elementary entities called elements. Each element is obtained by connecting the nodes and each node is associated with a set of generalized displacements and forces, depending on the kind of element.



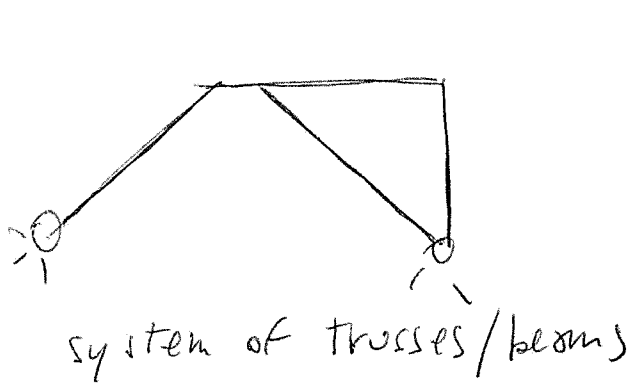
3D continuum divided
into smaller 3D elements



3D element

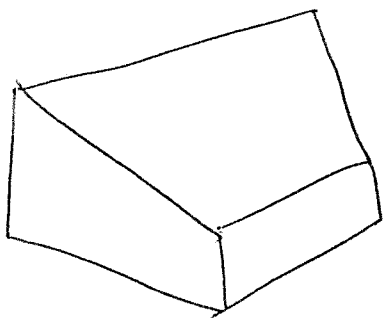
Considering the example in the figure, the element is composed of 8 nodes; each node is associated with three displacement components and three nodal force components. The relation between forces and displacement is obtained by referring to the FE theoretical framework discussed next.

The idea of dividing a structure into smaller elementary portions has been already introduced when analyzing systems of trusses and beams.

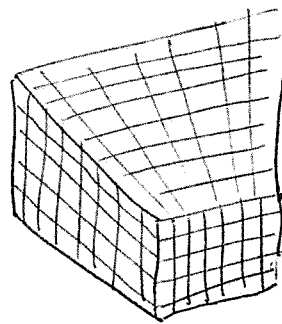


In this case the procedure for solving the linear static problem consists in assembling the governing

equations starting from the force-displacement relation available for each element. It can be useful to highlight that the force-displacement relation was derived by application of the PCVM. How can the approach be extended to a generic 2D and 3D case? How can the procedure be ~~developed~~ developed in a fully consistent displacement-based approach?



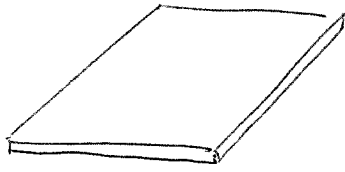
3D body



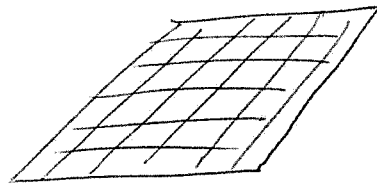
subdivision into small 3D elements

How to find force-displacement relations for the

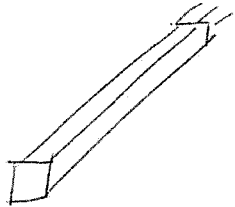
elements? How to assemble the contributions?



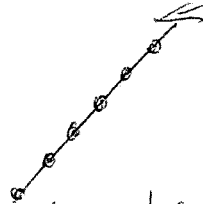
thin plate



thin plate FEM model



slender beam



slender beam FEM model

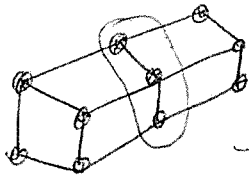
Irrespective on the kind of element, the FEM operates an idealization of the structure and allows to seek for an approximate solution, exactly as in the case of the Ritz and Galerkin methods; in a certain sense, the FEM can be easily seen as a Ritz-like approach where the trial functions are defined at local level (and not at global level as in Ritz).

To summarize, the method can be conceptually divided into the following steps:

1. Subdivision of the structure into small portions. This operation is the generation of the mesh, where:

mesh = nodes + elements

The various elements composing the mesh are connected each other in correspondence of the nodes.



the compatibility is enforced at nodal level. Along the frontier it is not necessarily satisfied (this depends upon the choice of the shape functions).

2. Definition of the elements to be used.

The elements could be 3D/2D/1D, depending on the modeling choices of the analyst.

In addition, once the dimensionality of the element is defined, one must choose which type of element to consider in terms of element formulation: linear interpolating functions? Or quadratic? Or cubic?

FEM codes typically offer a large library of elements that one should choose based on the needs of the analysis, its particularities, the experience with the code,...

3. Formulation of the equilibrium conditions

(if operating in a displacement-based approach) or, to a more general extent,

of the stationarity condition of the variational principle adopted,

For the scopes of this course the variational formulation is always intended in terms of Principle of Virtual Works or Total Potential Energy. Other formulations are clearly possible.

4. Solution of the discrete equations obtained from the previous step.

As in the case of the method of Ritz, the FEM allows to transform the continuum problem - which is governed by Partial Differential Equations (PDE) - into an algebraic problem.

In the context of a displacement-based formulation, the discrete equations express the equilibrium conditions at the nodes of the mesh.

It can be observed that the solution of the discrete equations leads to the satisfaction of the equilibrium in a weak sense.

The solution is indeed ~~an~~ approximate, and the equilibrium is not fulfilled locally.

5. Evaluation of the displacement field (using the shape functions) and recovery of stress/strain field.

Once the nodal displacements (i.e. the unknowns of the problem), the displacement field can be reconstructed in the overall domain by making use of the shape functions to interpolate the nodal values.

Formulation for the linear static case using 3D elements

Consider, as a starting point, the case of a 3D body (this is, as a matter of fact, the simplest case as no kinematic models are implied).

The Principle of Virtual Works is written as:

$$\int_V \delta \underline{\underline{\epsilon}} : \underline{\underline{\sigma}} dV = \int_V \delta \underline{s} \cdot \underline{f} dV + \int_A \delta \underline{s} \cdot \underline{f}^A dA + \sum_{i=1}^N \delta \underline{s}_i \cdot \underline{F}_i^c$$

where \underline{s}^T : displacement components = $\{u(x,y,z) \ v(x,y,z) \ w(x,y,z)\}$

\underline{f} : volume forces

\underline{f}^A : surface forces

\underline{F}_i^c : concentrated load

$\underline{\underline{\sigma}}, \underline{\underline{\epsilon}}$: stress/strain tensor

By organizing the tensor/vector components into matrices/vectors, it is possible to re-write the PVW as:

$$\int_V \delta \underline{\epsilon}^T \underline{\sigma} dV = \int_V \delta \underline{s}^T \underline{f} dV + \int_A \delta \underline{s}^T \underline{f}^A dA + \sum_{i=1}^N \delta \underline{s}_i^T \underline{F}_i^c$$

As in the case of Ritz, the displacement field is approximated as the product between

the shape functions and nodal displacements.

$$\underline{s} = \underline{s}(x, y, z) = \underline{N}(x, y, z) \underline{u}$$

where \underline{u} defines the vector of the nodal displacements (which are independent on x, y, z)

Accordingly the strain-displacement relation is found as:

$$\varepsilon_{ik} = \frac{1}{2} \left(u_{i/k} + u_{k/i} \right)$$



$$\varepsilon_{xx} = u/x$$

$$\varepsilon_{yy} = v/y$$

$$\varepsilon_{zz} = w/z$$

$$\gamma_{yz} = v/z + u/y$$

$$\gamma_{xz} = u/z + w/x$$

$$\gamma_{xy} = u/y + v/x$$

$$\Rightarrow \underline{\varepsilon} = \begin{pmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{yz} \\ \gamma_{xz} \\ \gamma_{xy} \end{pmatrix} = \begin{bmatrix} 1/x & 0 & 0 \\ 0 & 1/y & 0 \\ 0 & 0 & 1/z \\ 0 & 1/z & 1/y \\ 1/z & 0 & 1/x \\ 1/y & 1/x & 0 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}$$

or, in compact form:

$$\underline{\varepsilon} = \underline{D} \underline{s}, \text{ where } \underline{D} \text{ is a differential}$$

operator collecting the derivatives wrt x, y, z as illustrated above

Introducing now the approximation of the displacement field as $\underline{s} = \underline{N} \underline{u}$, it is possible to express the strains as function of the nodal displacements:

$$\underline{\epsilon} = \underline{\mathcal{D}} \underline{s} \\ = \underline{\mathcal{D}} \underline{N} \underline{u}$$

$$\boxed{\underline{\epsilon} = \underline{B} \underline{u}} \quad \text{where } \underline{B} = \underline{\mathcal{D}} \underline{N} \quad \text{is the matrix collecting the derivatives of the shape function.}$$

To express the PVW as function of the nodal displacements it is necessary to introduce the constitutive law. In the linear hyperelastic case it can be written as:

$$\boxed{\underline{\sigma} = \underline{C} \underline{\epsilon}} \quad \left(\text{which is the matrix representation of the tensor description} \right. \\ \left. \sigma_{in} = C_{inrs} \epsilon_{rs} \right)$$

The PVW can be reformulated as: (consider, for now, the single element)

$$\int_V \delta \underline{\epsilon}^T \underline{C} \underline{\epsilon} dV = \int_V \delta \underline{s}^T \underline{f} dV + \int_A \delta \underline{s}^T \underline{f}^A dA + \sum_{i=1}^N \delta \underline{s}_i^T \underline{F}_i^c$$

and recalling that:

$$\underline{s} = \underline{N} \underline{u} \\ \underline{\epsilon} = \underline{B} \underline{u}$$

$$\underline{u}^T \int_V \underline{B}^T \underline{C} \underline{B} dV \underline{u} = \underline{u}^T \int_V \underline{N}^T \underline{f} dV + \underline{u}^T \int_A \underline{N}^T \underline{f}^A dA + \\ + \sum_i \underline{u}^T \underline{N}_i^T \underline{F}_i^C$$

It is now recalled that the continuum is divided into an assembly of elements so:

$$\delta W_i = \sum_{e=1}^{Nel} \delta W_i^e \quad \text{in other words, total virtual work is the sum of the contributions of each element}$$

$$\delta W_e = \sum_{e=1}^{Nel} \delta W_e^e$$

And so:

$$\sum_{e=1}^{Nel} \underline{u}_e^T \int_{V_e} \underline{B}_e^T \underline{C}_e \underline{B}_e dV_e \underline{u}_e = \\ = \sum_{e=1}^{Nel} \underline{u}_e^T \int_{V_e} \underline{N}_e^T \underline{f}_e dV_e + \underline{u}_e^T \int_{A_e} \underline{N}_e^T \underline{f}_e^A dA_e + \\ \sum_{e=1}^{Nel} \sum_{i=1}^N \underline{u}_{e,i}^T \underline{F}_i^C$$

Or, with a more compact representation:

$$\sum_e \underline{u}_e^T \underline{K}_e \underline{u}_e = \sum_e \underline{u}_e^T \left(\underline{F}_e^V + \underline{F}_e^A + \sum_i \underline{F}_i^C \right)$$

where \underline{K}_e : stiffness matrix of the element

$\underline{F}_e^V, \underline{F}_e^A, \underline{F}_i^C$: volume, surface and concentrated forces associated with the element e .

The assembly procedure consists in expanding the contribution of each single element at global equation level. Formally one can write this operation as:

$$\underline{u}_e = \underline{\Omega}_e \underline{u} \quad \text{with } \underline{\Omega}_e \text{ boolean matrix of the element } e$$

So:

$$\underline{u}^T \left(\sum_e \underline{\Omega}_e^T \underline{k}_e \underline{\Omega}_e \right) \underline{u} = \underline{u}^T \left[\sum_e \underline{\Omega}_e^T \left(\underline{F}_e^V + \underline{F}_e^A + \sum_i \underline{F}_i^C \right) \right]$$

From which:

$$\underline{u}^T \underline{k} \underline{u} = \underline{u}^T \underline{F} \quad (\forall \underline{u})$$

$$\Rightarrow \underline{k} \underline{u} = \underline{F}$$

with \underline{k} : assembled stiffness matrix

\underline{F} : assembled force vector

having denoted:

$$\underline{k} = \sum_e \underline{\Omega}_e^T \underline{k}_e \underline{\Omega}_e \quad \text{and}$$

$$\underline{F} = \sum_e \underline{\Omega}_e^T \left(\underline{F}_e^V + \underline{F}_e^A + \sum_i \underline{F}_i^C \right)$$

It is important to recall that the assembly of the equations is never performed using the Boolean matrices \underline{B}_e , but is done by directly accumulating the contributions of the elements according to the positions specified by the vectors of pointers. In this case, it can be written that:

$$\underline{K} = \underset{e=1}{\overset{Nel}{A}} \underline{K}_e$$

$$\underline{F} = \underset{e=1}{\overset{Nel}{A}} \underline{F}_e^V + \underline{F}_e^A + \sum_i \underline{F}_i^C$$

where A is the operator indicating the assembly of the equations.

Remarks

1. The stiffness matrix of the structure is, in general, singular. As no constraints have been introduced so far, rigid body motions are possible.

Regarding the boundary conditions it is thus necessary to impose the essential ones.

This operation can be conducted by removing rows and columns associated with the constrained dofs.

Note that natural conditions does not need to be enforced. They are satisfied in a weak-form sense, as they are included in the variational principle.

2. It is useful to illustrate analogies and differences with respect to Ritz.

FEM

$$\underline{K} \underline{u} = \underline{F}$$

- Unknowns: nodal displacements (clear "physical" meaning)

RITZ

$$\underline{K} \underline{c} = \underline{F}$$

- Unknowns: Ritz amplitudes (Lagrangian coordinates)

• K: sparse

• Number of dofs:
high

• Convergence by:

a. reducing element size

b. increasing shape functions order

• K: high density

• Number of dofs:
low

• Convergence by:

a. increasing trial function order

3. The formulation has been restricted to the linear static case. The effect of dynamic forces can be easily included by writing the external work as:

$$\begin{aligned}\delta W_e^e &= - \int_V \delta \underline{s}^T \rho \underline{\ddot{s}} dV \\ &= - \int_V \delta \underline{u}^T \underline{N}^T \rho \underline{N} \underline{\ddot{u}} dV\end{aligned}$$

and so:

$$\begin{aligned}\delta W_e &= \sum_{e=1}^{Nel} \delta W_e^e \\ &= \sum_{e=1}^{Nel} - \delta \underline{u}_e^T \int_{V_e} \underline{N}_e^T \underline{N}_e \rho dV_e \underline{\ddot{u}}_e \\ &= \sum_{e=1}^{Nel} - \delta \underline{u}_e^T \underline{M}_e \underline{\ddot{u}}_e\end{aligned}$$

where: $\underline{M}_e = \int_{V_e} \underline{N}_e^T \underline{N}_e \rho dV_e$

The mass matrix of the assembled set of equations is then:

$$\underline{M} = \sum_{e=1}^{N_{el}} \underline{M}_e$$

The final system of equations is then:

$$\boxed{\underline{M} \ddot{\underline{u}} + \underline{K} \underline{u} = \underline{F}}$$

Stiffness matrix of some elements in physical coordinates

To clarify the main aspects of the FE procedure, it is useful to discuss how to derive the stiffness matrix for some simple finite elements.

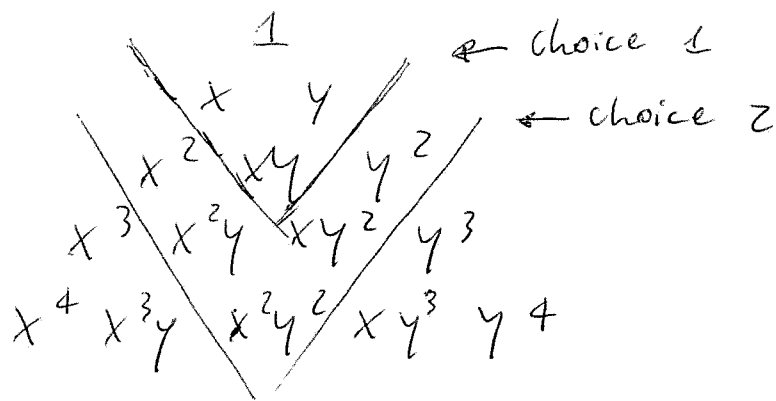
The operation is conducted here in the so-called physical coordinates, which are the coordinates where the problem is defined (an alternative procedure, discussed next, is based on the transformation of the problem in the "computational" domain).

The shape functions can be constructed by considering simple polynomial expansions (in FE, as opposed to Ritz, low order polynomials are generally used, thus the ill-conditioning issues associated with high-order functions are not generally encountered).

To this aim, the Pascal Triangle can be taken as a mean for selecting the basis.

$$\begin{array}{ccccccc} & & & & 1 & & \\ & & & x & & y & \\ & & x^2 & & xy & & y^2 \\ x^3 & & x^2y & & xy^2 & & y^3 \end{array}$$

where the terms are selected as reported below.



According to choice 1, 4 terms are considered (so the element is characterized by 4 nodes) and the expansion is complete up to the 1st order.

According to choice 2, 9 terms are considered (the element has 9 nodes) and the expansion is complete up to the 2nd order.

Note: Completeness of the expansion is achieved up to the order N if all of the terms of order N are retained as part of the expansion. For "choice 2" the first three rows are part of the expansion, whilst only a few elements are taken from the fourth and fifth lines of the triangle.

What reported above refers to the 2D case.

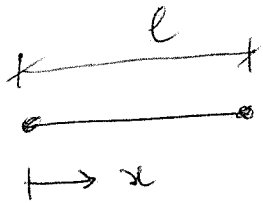
In the 1D case one can simply consider

a polynomial expansion in the form

$$1 \quad x \quad x^2 \quad \dots \quad x^n$$

Linear bar element

Consider a two-node bar element.



How to construct an interpolating scheme such that

$$u = \underline{N} \underline{u} \quad ?$$

$$\begin{array}{c} \uparrow \\ \text{node values} \\ \uparrow \\ u = u(x) \end{array}$$

As far as the element is made of two elements, the expansion is taken as:

$$\underline{X} = [1 \quad x]$$

And the displacement u is approximated as:

$$u = [1 \quad x] \begin{Bmatrix} \alpha_1 \\ \alpha_2 \end{Bmatrix} = \alpha_1 + \alpha_2 x$$

$$= \underline{X} \underline{\alpha} \quad (\text{this is the kind of strategy used in Ritz-like approaches so far})$$

In FE the idea is that of considering the nodal displacements as unknowns instead of the Lagrangian coordinates $\underline{\alpha}$.

It is then imposed:

$$\begin{aligned} u(0) &= u_1 \\ u(l) &= u_2 \end{aligned} \Rightarrow \begin{bmatrix} 1 & 0 \\ 1 & l \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \end{Bmatrix} = \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}$$

From which:

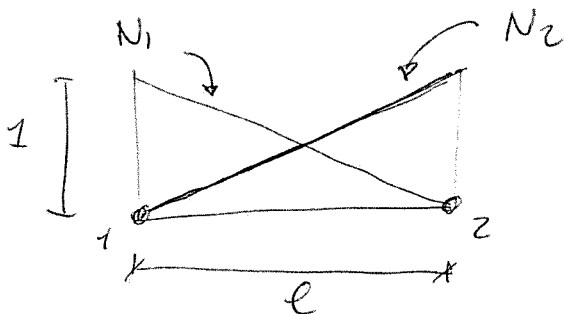
$$\begin{cases} \alpha_1 = u_1 \\ \alpha_2 = \frac{u_2 - u_1}{l} \end{cases}$$

leading to the following interpolating scheme:

$$\begin{aligned} u &= u_1 + \frac{u_2 - u_1}{l} x \\ &= \left(1 - \frac{x}{l}\right) u_1 + \frac{x}{l} u_2 \end{aligned}$$

$$u = \begin{bmatrix} 1 - \frac{x}{l} & \frac{x}{l} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \underline{\underline{N}} \underline{\underline{u}}$$

$1 \times 2 \quad 2 \times 1$



The shape functions are then linear functions of the coordinate x . They are equal to 1 in correspondence of the associated node and zero on the other node, i.e.

$$N_1 = 0 \text{ in } x = x_2; \quad N_1 = 1 \text{ in } x = x_1$$

$$N_2 = 0 \text{ in } x = x_1; \quad N_2 = 1 \text{ in } x = x_2$$

The stiffness matrix of the single element is available as;

$$\underline{\underline{K}} = \int_V \underline{\underline{B}}^T E \underline{\underline{B}} dV = EA \int_0^l \underline{\underline{B}}^T \underline{\underline{B}} dx$$

where

$$\epsilon_{xx} = u_{/x} = ()_{/x} u$$

$$= \underline{\underline{D}} u$$

and

$$u = [N_1 \quad N_2] \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \quad \text{so;}$$

$$\underline{\underline{B}} = \underline{\underline{D}} \underline{\underline{N}} = [N_{1/x} \quad N_{2/x}]$$

$$= [-1/l \quad 1/l]$$

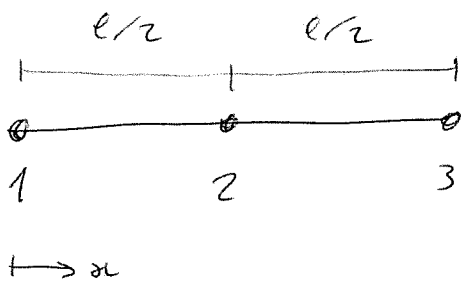
$$\underline{\underline{K}} = EA \int_0^l \begin{bmatrix} -1/l \\ 1/l \end{bmatrix} [-1/l \quad 1/l] dx$$

$$= \frac{EA}{l} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Quadratic bar element

One possibility for refining the accuracy of the predictions consists in increasing the order of the interpolation functions. Quadratic elements are widely used and an example is illustrated here with regard to a bar element.

The element is now made by three nodes, two external and one internal.



The expansion, due to the additional dof, is taken as:

$$u = \alpha_1 + \alpha_2 x + \alpha_3 x^2$$

Imposing

$$\begin{aligned} u(0) &= u_1 \\ u(l/2) &= u_2 \\ u(l) &= u_3 \end{aligned} \Rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 1 & l/2 & l^2/4 \\ 1 & l & l^2 \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{Bmatrix} = \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix}$$

From which:

$$\alpha_1 = u_1$$

$$\alpha_2 = \frac{1}{l} (-3u_1 + 4u_2 - u_3)$$

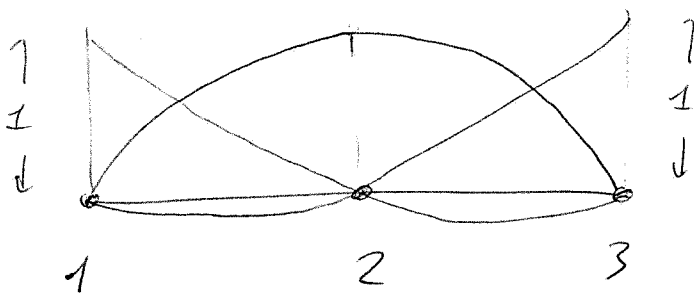
$$\alpha_3 = \frac{2}{l^2} (u_1 - 2u_2 + u_3)$$

The displacement field is then:

$$u = \left[\underbrace{1 - \frac{3x}{\ell} + \frac{2}{\ell^2} x^2}_{N_1} \mid \underbrace{\frac{4}{\ell} x - \frac{4}{\ell^2} x^2}_{N_2} \mid \underbrace{-\frac{x}{\ell} + \frac{2}{\ell^2} x^2}_{N_3} \right] \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix}$$

where the functions N_i are nothing but the lagrange polynomials $P_j = \prod_{j \neq k}^3 \frac{x_k - x}{x_k - x_j}$, i.e.

polynomial functions which are equal to 1 at the corresponding node and 0 at the other nodes.



$$N_i(x_j) = \delta_{ij}$$

The stiffness matrix is obtained as:

$$\underline{\underline{K}} = \int_0^\ell EA \underline{\underline{B}}^T \underline{\underline{B}} dx$$

with $\underline{\underline{B}} = \mathcal{D} \underline{\underline{N}}$ and $\mathcal{D} = ()_{,x}$

$$\text{so: } \underline{\underline{B}} = \left[-\frac{3}{\ell} + \frac{4}{\ell^2} x \mid \frac{4}{\ell} - \frac{8}{\ell^2} x \mid -\frac{1}{\ell} + \frac{4}{\ell^2} x \right]$$

From which:

$$\underline{k} = \frac{EA}{\ell} \begin{bmatrix} 7/3 & -8/3 & 1/3 \\ & 16/3 & -8/3 \\ & & 7/3 \end{bmatrix}$$

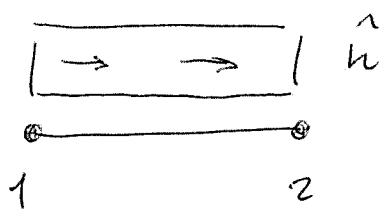
External loads

Any set of external loads has to be transformed into a set of nodal, discrete forces. As a matter of fact, the FEM discretization transforms the equilibrium conditions into a set of discrete equations expressing the equilibrium at node level. Accordingly, the forces (external loads) have to be translated into a set of nodal loads.

In the case of concentrated loads, which have to be applied in correspondence of nodes, this operation is clearly not necessary.

For distributed loads the operation is conducted by projecting the external loads over the shape functions of the FEM approximation.

Consider the case of a distributed load applied to a linear bar element



where

$$N_1 = 1 - \frac{x}{l}$$

$$N_2 = \frac{x}{l}$$

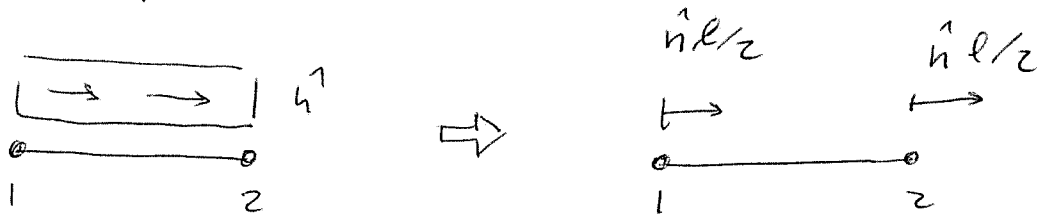
The discretization is automatically available after writing the external virtual work

$$\delta W_e = \int_0^l \delta u \, q \, dx$$

$$= \underline{f} \underline{u}^T \int_0^l \underline{N}^T \hat{h} dx$$

$$= \underline{f} \underline{u}^T \int_0^l \begin{bmatrix} 1-x/l \\ x/l \end{bmatrix} \hat{h} dx = \underline{f} \underline{u}^T \begin{bmatrix} \hat{h} l/2 \\ \hat{h} l/2 \end{bmatrix}$$

The load is then transformed into two nodal forces of magnitude $\hat{h} l/2$.



While this result is quite intuitive (due to the simplicity of the load and the element formulation) the approach is of general validity and can be considered for transforming any set of load into the corresponding conformal loads.

Consider now a quadratic bar element

$$N_1 = 1 - 3 \frac{x}{l} + 2 \frac{x^2}{l^2}$$

$$N_2 = 4 \frac{x}{l} - 4 \frac{x^2}{l^2}$$

$$N_3 = - \frac{x}{l} + 2 \frac{x^2}{l^2}$$

How to "transform" the distributed load into the conformal nodal forces?

Again, the external virtual work can be

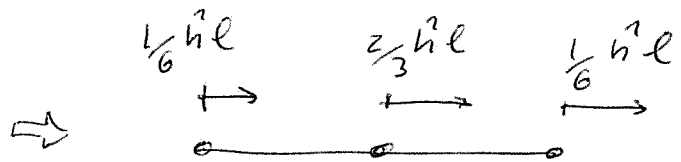
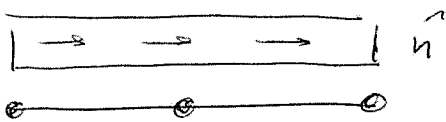
written as:

$$\delta W_e = \int_0^l \underline{f} u \hat{n} dx$$

$$= \underline{f} u^T \int_0^l \underline{N}^T \hat{n} dx = \underline{f} u^T \int_0^l \begin{bmatrix} 1 - 3\frac{x}{l} + 2\frac{x^2}{l^2} \\ 4\frac{x}{l} - 4\frac{x^2}{l^2} \\ -\frac{x}{l} + 2\frac{x^2}{l^2} \end{bmatrix} \hat{n} dx$$

$$= \underline{f} u^T \begin{bmatrix} \hat{n} l / 6 \\ 2\hat{n} l / 3 \\ \hat{n} l / 6 \end{bmatrix}$$

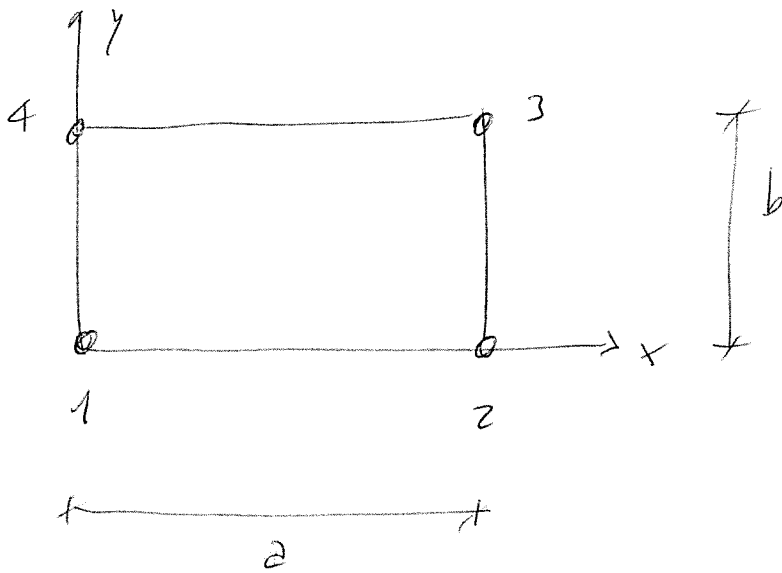
The loads are then reported at the nodes as:



Bilinear membrane element

Consider now a 2D element. For simplicity a membrane element (zero bending stiffness) is considered, so that no kinematic models need to be introduced and the number of dofs is restricted to the membrane ones.

The element is represented as:



The element is made of 4 nodes, so that a linear interpolation is adopted along the directions x and y .

Each node has two degrees of freedom, which are the displacements along x and y .

Recalling the Pascal Triangle

$$\begin{array}{c} 1 \\ x \quad y \\ x^2 \quad xy \quad y^2 \end{array} \rightarrow \underline{X} = [1 \quad x \quad y \quad xy]$$

Four polynomial terms are taken, leading to a basis complete up to the first order.

According to this choice, the displacement field is interpolated as:

$$u = [1 \quad x \quad y \quad xy] \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{Bmatrix}$$

$$v = [1 \quad x \quad y \quad xy] \begin{Bmatrix} \alpha_5 \\ \alpha_6 \\ \alpha_7 \\ \alpha_8 \end{Bmatrix}$$

or:

$$u = \underline{X}^T \underline{\alpha}_u$$

$$v = \underline{X}^T \underline{\alpha}_v$$

where:

$$\underline{\alpha}_u^T = \{ \alpha_1 \quad \alpha_2 \quad \alpha_3 \quad \alpha_4 \}$$

$$\underline{\alpha}_v^T = \{ \alpha_5 \quad \alpha_6 \quad \alpha_7 \quad \alpha_8 \}$$

The shape functions, can be built as done in the 1D case:

$$\begin{aligned} u(0,0) &= u_1 \\ u(a,0) &= u_2 \\ u(a,b) &= u_3 \\ u(0,b) &= u_4 \end{aligned} \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & a & 0 & 0 \\ 1 & a & b & ab \\ 1 & 0 & b & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}$$

$$\underline{\underline{A}}$$

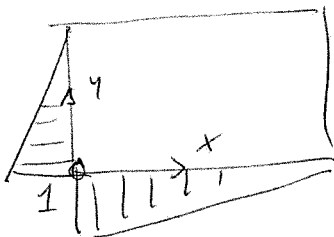
$$\underline{\underline{\alpha}}_u = \underline{\underline{u}}_u$$

From which: $\underline{\underline{\alpha}}_u = \underline{\underline{A}}^{-1} \underline{\underline{u}}_u$ and for:

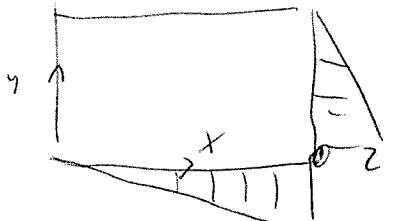
$$u = \underline{\underline{X}} \underline{\underline{\alpha}}_u = \underline{\underline{X}} \underline{\underline{A}}^{-1} \underline{\underline{u}}_u = \underline{\underline{N}} \underline{\underline{u}}_u$$

where: $\underline{\underline{N}} = \underline{\underline{X}} \underline{\underline{A}}^{-1}$

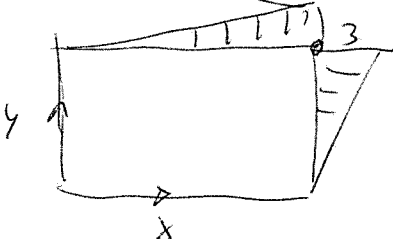
For a rectangular element it is obtained:



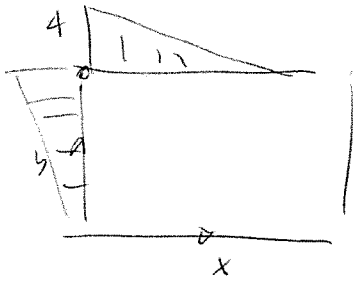
$$N_1 = \left(1 - \frac{x}{a}\right) \left(1 - \frac{y}{b}\right)$$



$$N_2 = \frac{x}{a} \left(1 - \frac{y}{b}\right)$$



$$N_3 = \frac{x}{a} \frac{y}{b}$$



$$N_4 = \left(1 - \frac{x}{a}\right) \frac{y}{b}$$

The evaluation of the stiffness matrix is now:

$$\underline{\underline{K}} = \int_A t \underline{\underline{B}}^T \underline{\underline{C}} \underline{\underline{B}} dA$$

(Recall that $\underline{\underline{K}}$ is available from the internal

$$\text{work: } \delta W_i = \int_V \delta \underline{\underline{\epsilon}}^T \underline{\underline{\sigma}} dV = \int_V \delta \underline{\underline{\epsilon}}^T \underline{\underline{C}} \underline{\underline{\epsilon}} dV$$

and, for a plane problem:

$$\delta W_i = \int_A \delta \underline{\underline{\epsilon}}^T \underline{\underline{C}} t \underline{\underline{\epsilon}} dV = \delta \underline{\underline{u}}^T \int_A \underline{\underline{B}}^T \underline{\underline{C}} t \underline{\underline{B}} dA$$

where:

$$\underline{\underline{C}} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \quad (\text{plane stress})$$

and

$$\begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} 1/x & 0 \\ 0 & 1/y \\ 1/y & 1/x \end{bmatrix} \begin{Bmatrix} u \\ v \end{Bmatrix} \Rightarrow \underline{\underline{D}} = \begin{bmatrix} 1/x & 0 \\ 0 & 1/y \\ 1/y & 1/x \end{bmatrix}$$

and:

$$\underline{B} = \underline{D} \underline{N} = \begin{bmatrix} N_{1/x} & N_{2/x} & N_{3/x} & N_{4/x} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & N_{1/y} & N_{2/y} & N_{3/y} & N_{4/y} \\ N_{2/y} & N_{1/y} & N_{3/y} & N_{4/y} & N_{1/x} & N_{2/x} & N_{3/x} & N_{4/x} \end{bmatrix}$$

Despite the simplicity of the approach for constructing the shape functions, this approach is quite inefficient as demands for the evaluation of the trial functions element by element. Each element composing the structure is, in principle, associated with a different matrix A to be inverted.

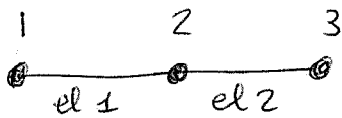
This is one of the motivations for considering parametric elements, a class of finite elements where the shape functions are defined on a domain which is the same for all the elements, the computational domain.

Assembly of the equations (and compatibility conditions)

The assembly procedure is conducted by summing up the individual (= of each element) contributions at global level. The operation is performed by placing the stiffness contributions of the elements in specific portions in the global stiffness matrix (by making use of the vectors of pointers).

Compatibility of the displacements at the common nodes is implicitly enforced when assembling the equations.

Consider the two-element FEM model below:



The model is composed of two elements with two nodes (linear bar elements)

The stiffness matrix of the two elements reads:

$$\underline{K}^{(1)} = \frac{EA_1}{l_1} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} ; \quad \underline{K}^{(2)} = \frac{EA_2}{l_2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

and with a graphical representation:

$$= \begin{bmatrix} \square & \square \\ \square & \square \end{bmatrix} = \begin{bmatrix} \times & \times \\ \times & \times \end{bmatrix}$$

If one considers the two elements as separate entities, not connected each other in correspondence of the node 2, the assembled stiffness matrix of the structure would be:

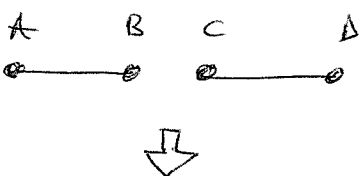
$$\begin{array}{c}
 \begin{array}{cccc}
 A & B & C & D \\
 \bullet & \bullet & \bullet & \bullet \\
 \hline
 & \text{el 1} & & \text{el 2}
 \end{array}
 \end{array}
 \Rightarrow \underline{\underline{K}} = \begin{bmatrix}
 \square & \square & & \\
 \square & \square & & \\
 & & \times & \times \\
 & & \times & \times
 \end{bmatrix}$$

However, this is not the case and the two elements share the same node 2. It follows that the compatibility of the nodal displacement is obtained by enforcing:

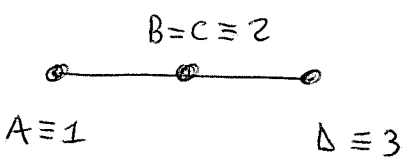
$$u_B = u_C$$

Note that this operation is particularly simple due to the fact that the unknowns of the problem are the displacements of the nodes. In the context of a Ritz based procedure, where the unknowns are given by the amplitudes of the trial functions, it is much more challenging to enforce the continuity of the displacements at the frontier.

After imposing $U_B = U_C$, the assembled stiffness matrix is obtained as illustrated below:

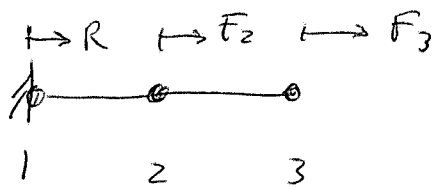


$B = C \equiv 2$



$$\Rightarrow K = \begin{bmatrix} \square & \square & \\ \square & \boxed{\times} & \times \\ & \times & \times \end{bmatrix}$$

Considering now the problem of a constrained structure (in the previous case no constraints were introduced and the resulting stiffness matrix was then singular)



The assembled set of equations is:

$$\begin{bmatrix} \frac{EA_1}{l_1} & -\frac{EA_1}{l_1} & \\ -\frac{EA_1}{l_1} & \frac{EA_1}{l_1} + \frac{EA_2}{l_2} & -\frac{EA_2}{l_2} \\ & -\frac{EA_2}{l_2} & \frac{EA_2}{l_2} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} R \\ F_2 \\ F_2 \end{Bmatrix}$$

The compatibility condition to be enforced, at global equation level, regards the essential condition $u_1 = 0$.

Again, this can be easily done as the prescribed displacement is associated with one single unknown of the problem, which is then removed from the set of solving equations.

$$\begin{bmatrix} \frac{EA_1}{\ell_1} & -\frac{EA_1}{\ell_1} & 0 \\ -\frac{EA_1}{\ell_1} & \frac{EA_1}{\ell_1} + \frac{EA_2}{\ell_2} & -\frac{EA_2}{\ell_2} \\ 0 & -\frac{EA_2}{\ell_2} & \frac{EA_2}{\ell_2} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} R \\ F_2 \\ F_3 \end{Bmatrix}$$

(Note that the reaction force R , which is an unknown of the problem, does not enter the first set of equations. If needed it can be evaluated once u_2 and u_3 are available)

Other strategies can be implemented for imposing the boundary conditions

1. Lagrange multipliers

The approach consists in "enriching" the variational statement by enforcing the constraint using Lagrange multipliers

Assuming that the constraint is linear, i.e.

it can be expressed in the generic form

$$\underline{A} \underline{u} = \underline{b}$$

Then the total potential energy is written as:

$$\Pi = \frac{1}{2} \underline{u}^T \underline{K} \underline{u} - \underline{u}^T \underline{f} + \underline{\lambda}^T (\underline{A} \underline{u} - \underline{b})$$

And so:

$$\delta \Pi = \delta \underline{u}^T (\underline{K} \underline{u} - \underline{f}) + \delta \underline{\lambda}^T (\underline{A} \underline{u} - \underline{b}) + \delta \underline{u}^T \underline{A}^T \underline{\lambda}$$

From which the stationarity conditions are:

$$\begin{bmatrix} \underline{K} & \underline{A}^T \\ \underline{A} & \underline{0} \end{bmatrix} \begin{Bmatrix} \underline{u} \\ \underline{\lambda} \end{Bmatrix} = \begin{Bmatrix} \underline{f} \\ \underline{b} \end{Bmatrix}$$

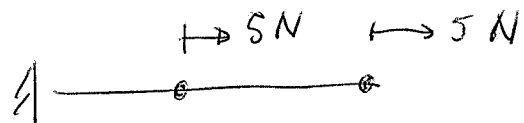
With reference to the previous example:

$$\underline{A} \underline{u} = \underline{b} \rightarrow [1 \ 0 \ 0] \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = 0$$

Consider, for simplicity, the case where:

$$\frac{EA_1}{l_1} = \frac{EA_2}{l_2} = 10 \text{ N/mm} \quad \text{and}$$

$$F_2 = F_3 = 5 \text{ N}$$



The discrete system is then:

$$\begin{bmatrix} 10 & -10 & 0 & 1 \\ -10 & 20 & -10 & 0 \\ 0 & -10 & 10 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ \lambda \end{Bmatrix} = \begin{Bmatrix} 0 \\ 5 \\ 5 \\ 0 \end{Bmatrix}$$

The solution of the linear system leads to:

$$u_1 = 0.0 \text{ mm}$$

$$u_2 = 1.0 \text{ mm}$$

$$u_3 = 1.5 \text{ mm}$$

$$\lambda = 10 \text{ N}$$

The result illustrates that the augmented system can be solved (K becomes non singular as far as the constraint is accounted for by the Lagrange multiplier) and, obviously, the imposed boundary condition $u_1 = 0$ is obtained as part of the solution.

It is interesting to note that the magnitude of the Lagrange multiplier λ is equal to the magnitude of the reaction force, in this case $5 + 5 \text{ N} = 10 \text{ N}$.

2. Penalty terms

Penalty formulations are commonly used in the context of Ritz-based procedure, as they facilitate the fulfillment of the boundary conditions (they make it possible the adoption of trial functions not respectful of the essential condition of the problem).

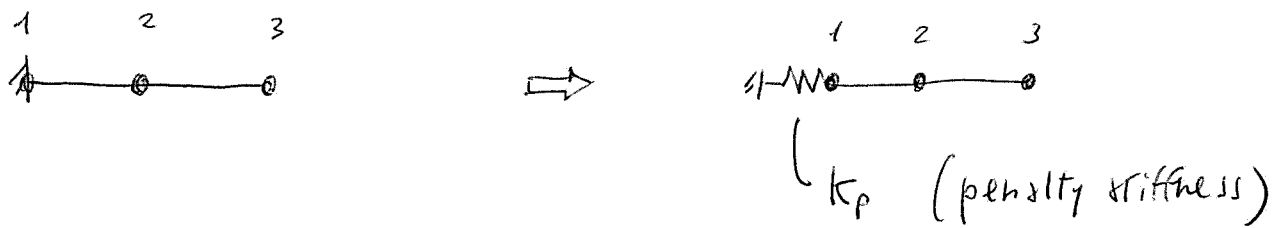
The same approach can be adopted in the context of finite element procedures.

The idea of a penalty approach consists in replacing the ideal constraint with an "artificial" spring with

a very high stiffness value (in relation to the stiffnesses of the problem under investigation).

The ideal case is recovered when the artificial stiffness is infinite. In practice this value should be taken as high as possible, with an upper bound dictated by the possible onset of ill-conditioning of the stiffness matrix.

For the example considered before:



The presence of the penalty stiffness determine the onset of an additional term to the strain energy in the form:

$$U_p = \frac{1}{2} k_p u_1^2 \quad \text{and so:}$$

$$\Pi = \Pi_{\text{old}} + \frac{1}{2} k_p u_1^2 \quad \left(\text{with } \Pi_{\text{old}} = \text{potential energy of the system without penalty spring} \right)$$

And so an additional contribution is obtained in the discrete equations, in this case in position (1,1). Indeed:

$$\delta \Pi = \delta \Pi_{\text{old}} + \delta U_p$$

$$\text{and } \delta U_p = \delta u_1 k_p u_1 \quad \leftarrow \text{ novel contribution}$$

Considering again

$$\frac{EA_1}{l_1} = \frac{EA_2}{l_2} = 10 \text{ N/mm} ; \quad F_2 = F_3 = 5 \text{ N}$$

One can assume a value for k_p equal to 100 times the stiffness of the bars, so;

$$k_p = 1000 \text{ N/mm};$$

The solving equations are then:

$$\begin{bmatrix} 10+1000 & -10 & 0 \\ -10 & 20 & -10 \\ 0 & -10 & 10 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 5 \\ 5 \end{bmatrix}$$

From which:

$$u_1 = 0.01 \text{ mm}$$

$$u_2 = 1.01 \text{ mm}$$

$$u_3 = 1.51 \text{ mm}$$

The results are very close to those obtained before. Note that the displacements are slightly higher as for as a finite amount of stiffness is now associated with the constraint of node 1.

As for as k_p is increased, the displacements tend asymptotically to the previous solution.

Parametric formulation

The approach in physical coordinates is intuitive and can be seen, in a certain sense, the direct transformation of the method of Ritz into a finite element counterpart.

However three (or less) main issues are worth of discussion

1. The procedure demands for the evaluation of the shape functions for any individual element composing the mesh.

The shape functions were obtained as:

$$\underline{N} = \underline{X} \underline{A}^{-1}$$

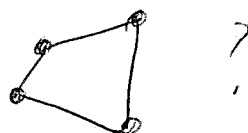
This operation has to be conducted for each element, which is a relatively costly operation.

2. Whenever the element is not regular, i.e. is not a square/cube in 2D/3D, one should understand how to perform the integrals for the evaluation of the stiffness matrix.

In this sense, the approach outlined does not seem particularly adequate to handle distorted elements

$$\underline{K} = \int_A \underline{B}^T \underline{C} \underline{B} \, dA$$

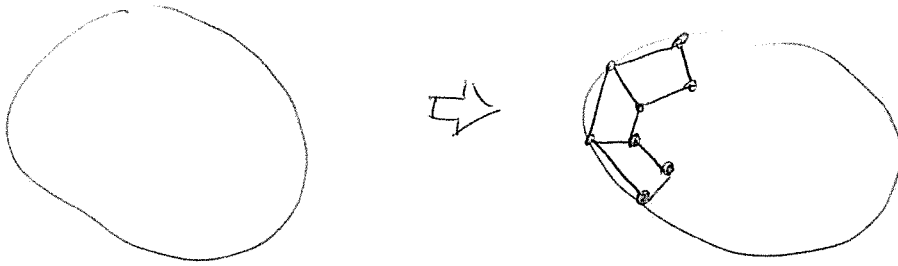
if the element is



How to integrate this expression

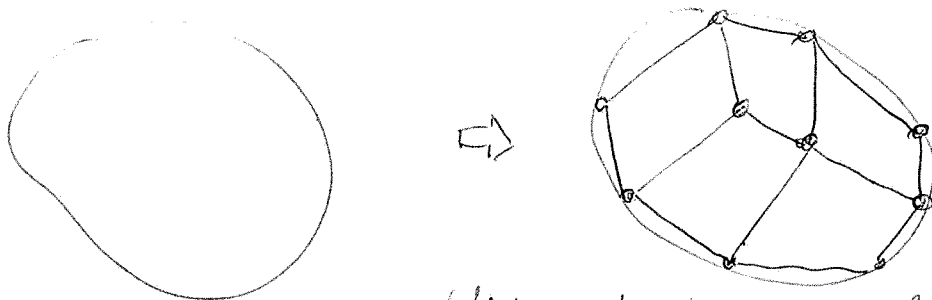
3. The method does not apply, in a natural way, to the modelling of non-straight edges.

Consider this case:



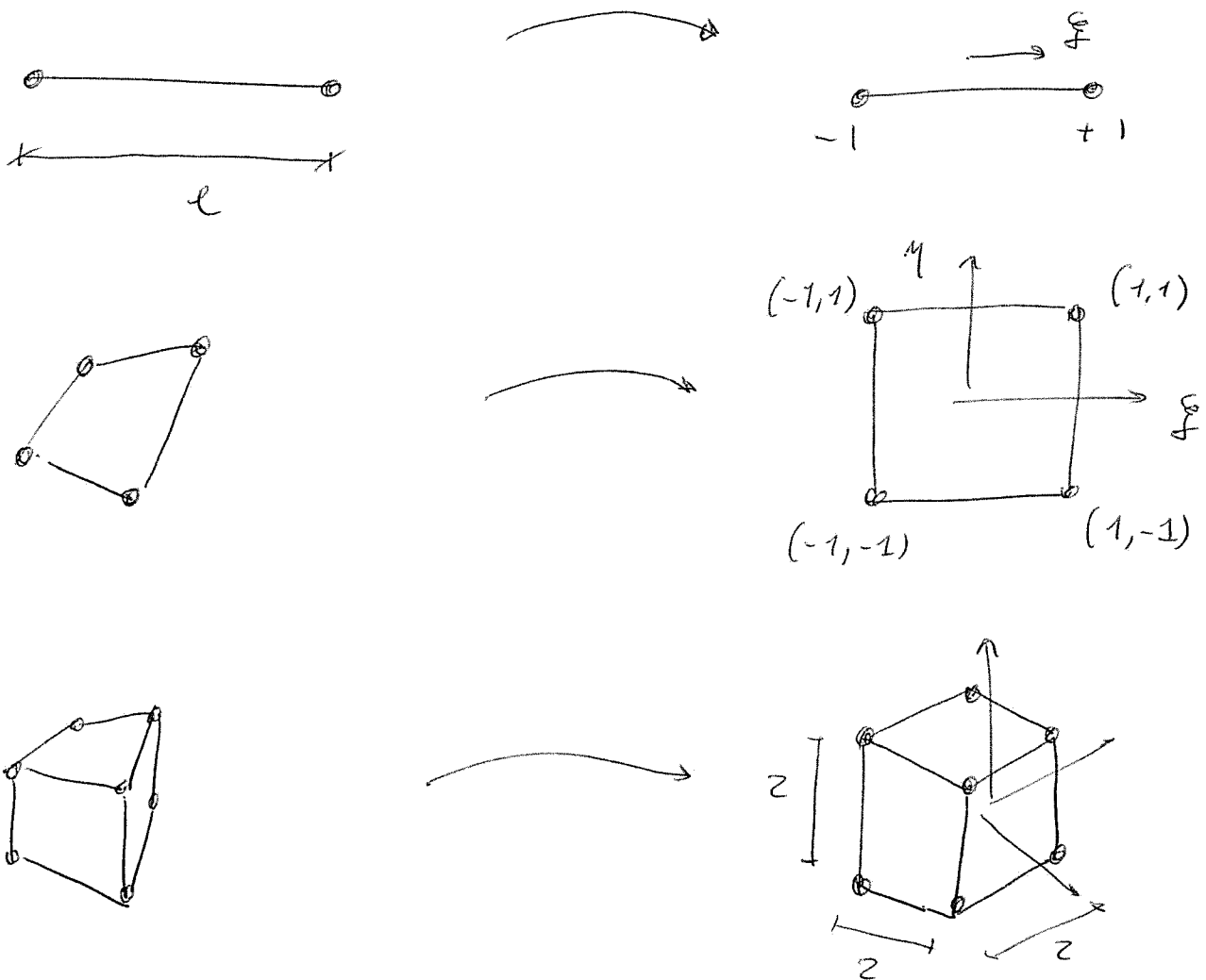
Curved edges are approximated as straight. Clearly the approximation becomes better and better as the size of the element is reduced. However one may want to consider not too small elements just for approximating the boundary in a proper way.

A typical case is given by the use of elements with an interpolation scheme of order higher than one. In this case one expects that less elements (and so larger) are needed, so a proper description of the edges is important.

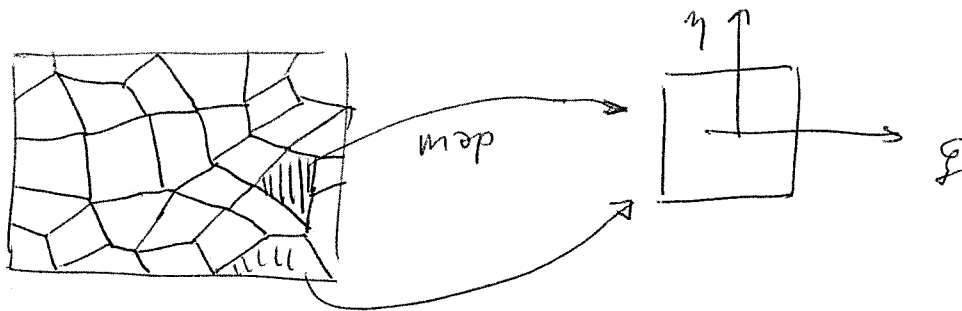


Using high order elements (larger), the approximation of curved edges can be very poor

The idea of the parametric formulation consists in operating a mapping from the physical domain into a reference (or computational) one. In other words the elements are transformed into a reference element which is equal to itself, irrespective of the shape of the element in the physical domain.



Consider a generic 2D mesh



each element, in turn, is mapped in the computational domain.

The consequences are that:

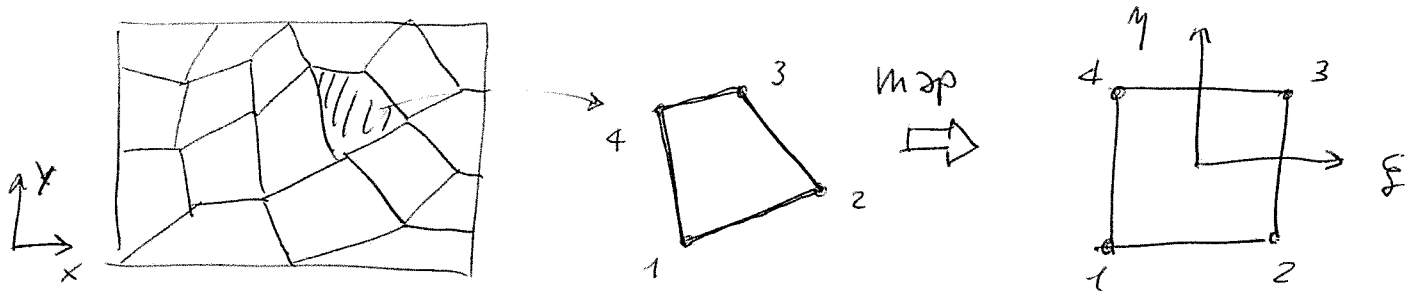
1. The element distortion should be such that the transformation is possible, meaning that it has to be biunivogue.
2. The shape functions are now defined in the computational domain, so they do not need to be evaluated for each single element.

The mapping from the physical to the computational domain can be written as:

$$\left. \begin{aligned} x &= f(\xi) && \text{in 1D} \\ \begin{Bmatrix} x \\ y \end{Bmatrix} &= \underline{f}(\xi, \eta) && \text{in 2D} \\ \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} &= \underline{f}(\xi, \eta, \zeta) && \text{in 3D} \end{aligned} \right\} \text{in general } \underline{x} = \underline{f}(\underline{\xi})$$

Isoparametric membrane element

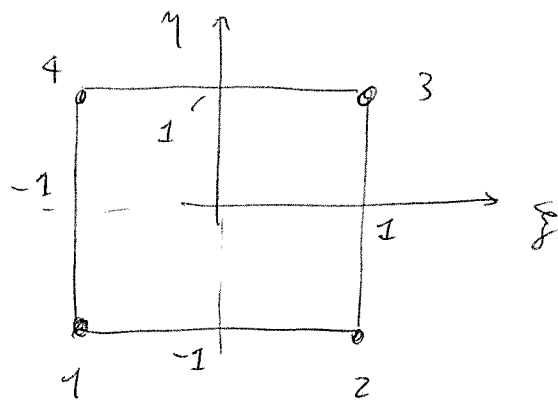
Consider a 2D element (no bending stiffness \Rightarrow membrane element) whose nodal coordinates are defined in the system x, y . The element is generically deformed



How to map it into the computational domain?

The idea is that of interpolating the element geometry, starting from the nodal coordinates, by make use of interpolation functions. Whenever these functions are taken as the shape functions used for interpolating the displacement field, the formulation is said to be isoparametric (if the order of the interpolation of the geometry is larger than the order used for the displacement field, it is said superparametric; if the order of the geometry interpolation is lower it is said subparametric)

When referring to the bi-unitary reference element, the definition is straightforward:



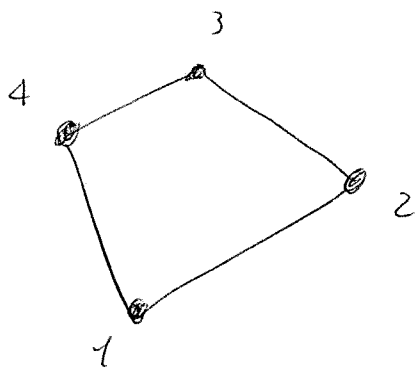
$$N_1 = \frac{1}{4} (1 - \xi) (1 - \eta)$$

$$N_2 = \frac{1}{4} (1 + \xi) (1 - \eta)$$

$$N_3 = \frac{1}{4} (1 + \xi) (1 + \eta)$$

$$N_4 = \frac{1}{4} (1 - \xi) (1 + \eta)$$

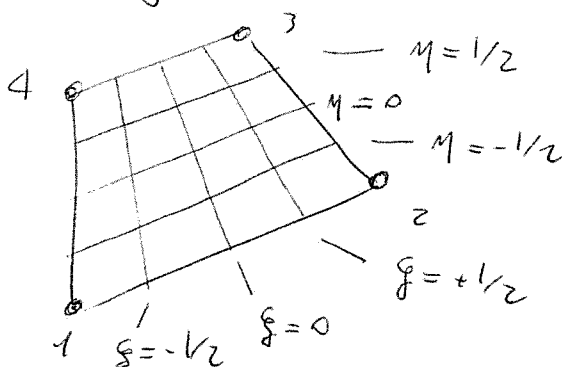
Accordingly, the mapping of the generic element is performed as:



$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{bmatrix} N_1 & N_2 & N_3 & N_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & N_1 & N_2 & N_3 & N_4 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{Bmatrix}$$

where (x_i, y_i) are the coordinates of the i -th node in the physical domain. Note that $N_i = N_i(\xi, \eta)$

so the relation above is indeed a transformation in the form $\underline{x} = \underline{f}(\underline{\xi})$. From a geometric standpoint the mapping happens as illustrated below:



It is now necessary to understand how to evaluate the derivatives of the shape functions, which are defined in (ξ, η) , with respect to the coordinates (x, y) , as required for the evaluation of the stiffness matrix.

Recall that:

$$\underline{K} = \int_A \underline{B}^T \underline{C} \underline{B} dA \quad \text{where}$$

$$\underline{B} = \underline{D} \underline{N} \quad \text{and} \quad \underline{\epsilon} = \underline{D} \underline{u} \quad \text{and:}$$

$$\underline{\epsilon} = \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \underbrace{\begin{bmatrix} 1/x & & & \\ & 1/y & & \\ & & 1/y & 1/x \end{bmatrix}}_{\underline{D}} \underbrace{\begin{Bmatrix} u \\ v \end{Bmatrix}}_{\underline{u}}$$

$$= \underbrace{\begin{bmatrix} 1/x & & & \\ & 1/y & & \\ & & 1/y & 1/x \end{bmatrix} \begin{bmatrix} N_1 & N_2 & N_3 & N_4 & & \\ & & & & N_1 & N_2 & N_3 & N_4 \end{bmatrix}}_{\underline{B}} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{Bmatrix}$$

As seen the evaluation of \underline{B} requires the ability to evaluate $N_{i/x}$ and $N_{i/y}$.

This operation can be conducted by observing that:

$$\frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial \xi}$$

$$\frac{\partial N_i}{\partial \eta} = \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial \eta}$$

Or, in matrix form:

$$\begin{Bmatrix} N_{i/g} \\ N_{i/y} \end{Bmatrix} = \begin{bmatrix} X/g & Y/g \\ X/y & Y/y \end{bmatrix} \begin{Bmatrix} N_{i/x} \\ N_{i/y} \end{Bmatrix}$$

$$= [J] \begin{Bmatrix} N_{i/x} \\ N_{i/y} \end{Bmatrix}$$

with $[J]$ = Jacobian of the transformation

It follows that:

$$\begin{Bmatrix} N_{i/x} \\ N_{i/y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} N_{i/g} \\ N_{i/y} \end{Bmatrix}$$

How to construct the Jacobian $[J]$?

$$[J] = \begin{bmatrix} X/g & Y/g \\ X/y & Y/y \end{bmatrix} = \begin{bmatrix} N_{1/g} & N_{2/g} & N_{3/g} & N_{4/g} \\ N_{1/y} & N_{2/y} & N_{3/y} & N_{4/y} \end{bmatrix} \begin{Bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{Bmatrix}$$

The evaluation of the matrix \underline{B} (necessary for calculating \underline{K}) is then performed with the following steps:

1. Construct $[J]$
2. Evaluate $N_{i/x}$ and $N_{i/y}$
3. Determine \underline{B} as:

$$\underline{B} = \begin{bmatrix} 1/x & 0 \\ 0 & 1/y \\ 1/y & 1/x \end{bmatrix} \begin{bmatrix} N_1 & N_2 & N_3 & N_4 & & \\ & & & & N_1 & N_2 & N_3 & N_4 \end{bmatrix} =$$

$$= \begin{bmatrix} N_{1/x} & N_{2/x} & N_{3/x} & N_{4/x} & & & & \\ & & & & N_{1/y} & N_{2/y} & N_{3/y} & N_{4/y} \\ N_{1/y} & N_{2/y} & N_{3/y} & N_{4/y} & N_{2/x} & N_{2/x} & N_{3/x} & N_{4/x} \end{bmatrix}$$

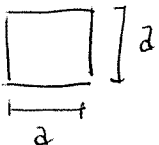
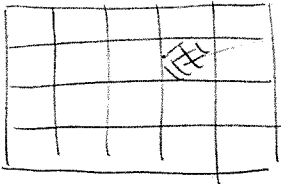
The stiffness matrix can be finally obtained as:

$$\left| \begin{aligned} \underline{K} &= \int_A t \underline{B}^T \underline{C} \underline{B} dA \\ &= \int_{-1}^{+1} \int_{-1}^{+1} t \underline{B}^T \underline{C} \underline{B} \det \underline{J} d\xi dy \end{aligned} \right|$$

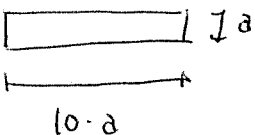
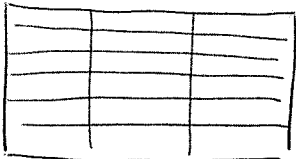
A few remarks

In general, it is a good practice to create a mesh as regular as possible.

In the ideal case of square element:


$$\underline{J} = \begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix} \left(= \begin{bmatrix} x/\xi & y/\xi \\ x/\eta & y/\eta \end{bmatrix} \right)$$

If the element is stretched along one direction:


$$\Rightarrow \underline{J} = \begin{bmatrix} 10a & 0 \\ 0 & a \end{bmatrix}$$

It is then clear that the diagonal elements provide a measure of the stretching of the element. More specifically, the ratio of the diagonal elements provides an indication of the stretching.

As a rule of thumb, the element can properly work up to a stretching of ≈ 10 . Higher values can lead to numerical issues.

Another interesting indication regarding the element shape is provided by the extra-diagonal terms, i.e.

y/ξ and x/η : they provide a measure of the distortion of the element. High values (of their ratio) indicate a severe distortion, which

has to be avoided.

Another measure for checking the entity of the element deformation is given by $\det \underline{J}$. Indeed:

$$dA = \det \underline{J} \, d\xi \, d\eta \quad \Rightarrow \quad \boxed{\det \underline{J} = \frac{dA}{d\xi \, d\eta}}$$

which is the ratio between the area of the infinitesimal element in the physical domain and in the computational one.

Numerical integration

The isoparametric formulation leads to the necessity of integrating numerically the expression of the stiffness matrix (and distributed loads, if any).

Indeed a closed-form integration can be performed in simple cases only. When the element is geometrically distorted the only viable option is the numerical integration.

The expression of \underline{k} was found as:

$$\underline{k} = t \int_{-1}^{+1} \int_{-1}^{+1} \underline{B}^T(\xi, \eta) \underline{C} \underline{B}(\xi, \eta) \det \underline{J} d\xi d\eta$$

The expression to be integrated is not even polynomial if the element is distorted.

The simplest approach to perform the integral consists in using a Gauss-Legendre technique.

For a generic function $f = f(\xi)$, the numerical integration is carried out as:

$$\int_{-1}^{+1} f(\xi) d\xi \approx \sum_{i=1}^{N_t} f(\xi_i) w_i$$

where ξ_i : Gauss points

w_i : Gauss weights

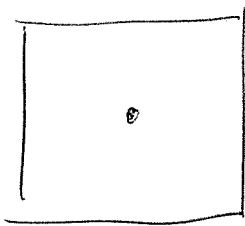
In two dimensions the integration is then:

$$\int_{-1}^{+1} \int_{-1}^{+1} f(\xi, \eta) d\xi d\eta = \sum_{i=1}^{N_{\xi}^i} \sum_{j=1}^{N_{\eta}^j} f(\xi_i, \eta_j) w_i w_j$$

where N_{ξ}^i and N_{η}^j are the number of Gauss points along ξ and η , respectively,

In most cases $N_{\xi}^i = N_{\eta}^j$. (for a generic element there is no reason for using more points along one direction with respect to the other).

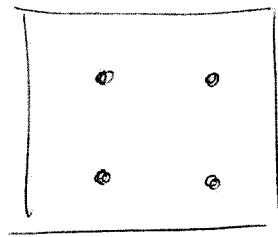
Two examples are reported below:



1-point rule

$$\xi_1 = \eta_1 = 0$$

$$w = 2$$



2-point rule

$$\xi_i = \pm 1/\sqrt{3}$$

$$\eta_i = \pm 1/\sqrt{3}$$

$$w_i = 1$$

The Gauss-Legendre rule allows to integrate exactly a polynomial expression of order $2N-1$, where N is number of Gauss points.

This means that a two-point rule ($N=2$) allows for the exact integration of a cubic polynomial.

Recall that, for a generically distorted element, the expression to be integrated is non polynomial.

How to choose the number of points?

- Many points \rightarrow accurate integration, but high cost to evaluate \underline{k}
- Few points \rightarrow low cost to build \underline{k} but possible inaccuracies

It is commonly defined:

1. Full integration: an integration aimed at integrating exactly the reference element (an element not distorted)
In the case of a bilinear membrane it would be $N=2$ (in order to integrate exactly a quadratic polynomial)
2. Reduced integration: an integration with less points with respect to the full one.

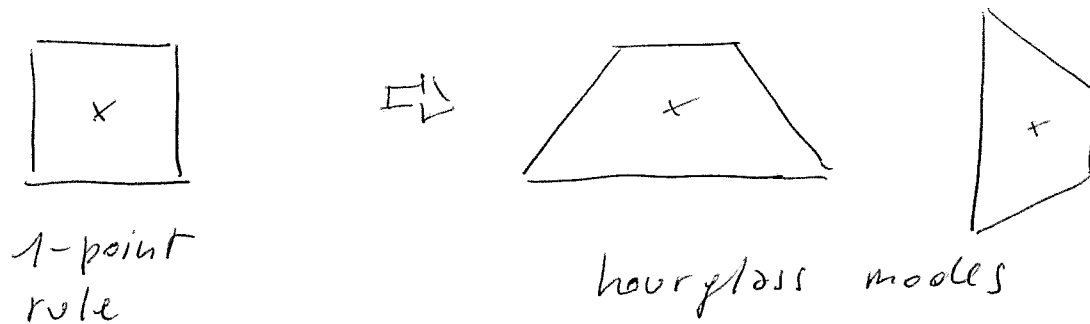
A reduced integration, typically with one point less than a full integration, can be a good choice. It offers the advantage of requiring less operations and, in addition, the error

associated with the relatively "poor" integration scheme tends to compensate the inherent over stiffness of a discrete model.

When choosing the integration rule it is always necessary to consider issues of ill-conditioning of \underline{K} , numerical instabilities or hourglass phenomena.

These latter are spurious deformation modes made possible by a lack of integration points.

For a 2D element integrated with 1 point rule



The two deformation modes in the figure are possible with a zero deformation energy. It may happen that the results are characterized by the presence of many elements exhibiting the typical hourglass mode. In these cases it is necessary to increase the order of the integration.

Recovery of stresses

In the linear elastic case the discrete problem is defined in the form:

$$\underline{K} \underline{u} = \underline{F}$$

Once the linear system is solved, the displacements \underline{u} are available at the nodes of the elements.

It is then possible to evaluate

1. Strains

$$\underline{\epsilon} = \underline{B} \underline{u}$$

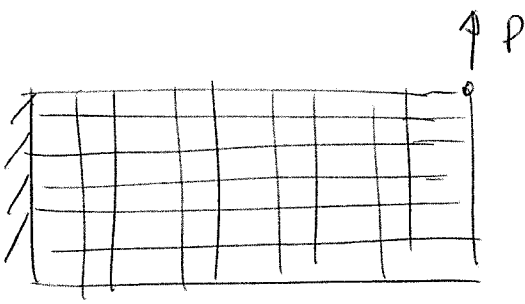
2. Stresses

$$\underline{\sigma} = \underline{C} \underline{B} \underline{u}$$

Recalling now that $\underline{B} = \underline{B}(\xi, \eta)$ it is necessary to establish how to check the stress levels; in other words, which points ξ_i, η_i one should consider?

From a practical standpoint one may be interested in the outer positions (the nodes) as they can be associated with the highest stress levels.

Consider, for instance, the structure in the figure



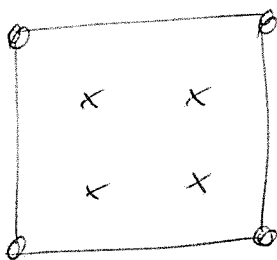
The bending-induced stresses will be highest at the top and the bottom.

Unfortunately the evaluation of stresses at the nodes leads to poor results. The quality of the stresses at the nodes is low.

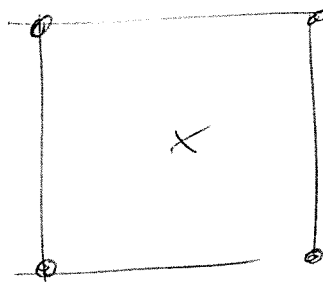
On the contrary, the stresses are nicely captured in correspondence of

- integration points
- Barlow points (integration points of the rule of order $N-1$, where N is the order of the rule considered for evaluating \underline{k})

If a 2-point rule is considered:



Integration points



Barlow point

The stresses can be evaluated in these points and extrapolated in any other point using a proper interpolation rule.

If considering the points of a 2-point integration rule:

$$\sigma_{xx} = \sigma_1 + \sigma_2 \xi + \sigma_3 \eta + \sigma_4 \xi \eta$$

where the coefficients σ_i have to be determined in order to interpolate the values in correspondence of the integration points.

Considering, as an example, the component σ_{xx} :

$$\begin{Bmatrix} \sigma_{xx}(\xi_1, \eta_1) \\ \sigma_{xx}(\xi_1, \eta_2) \\ \sigma_{xx}(\xi_2, \eta_1) \\ \sigma_{xx}(\xi_2, \eta_2) \end{Bmatrix} = \begin{bmatrix} 1 & \xi_1 & \eta_1 & \xi_1 \eta_1 \\ 1 & \xi_1 & \eta_2 & \xi_1 \eta_2 \\ 1 & \xi_2 & \eta_1 & \xi_2 \eta_1 \\ 1 & \xi_2 & \eta_2 & \xi_2 \eta_2 \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \end{Bmatrix}$$

$$\underline{\sigma}^{ip} = \underline{A} \underline{\sigma}, \text{ from which}$$

$$\underline{\sigma} = \underline{A}^{-1} \underline{\sigma}^{ip}$$

Once the coefficients $\underline{\sigma}$ are available, it is possible to determine the stress in any point as:

$$\sigma_{xx} = \sigma_1 + \sigma_2 \xi + \sigma_3 \eta + \sigma_4 \xi \eta$$

A few remarks

The FEM can be seen as a special case of the method of Ritz. It follows that the same considerations apply in terms of convergence of the solution.

In this sense, for a displacement-based finite element model:

1. The total potential energy Π is higher with respect to the exact case
- (2. The strain energy U is lower with respect to the exact case)
3. The total potential energy Π gets smaller and smaller when the mesh is progressively refined
4. The strain energy U increases as the mesh is refined.

These conclusions hold if

1. The elements are compatible

This means that compatibility should be guaranteed both internally (the use of continuous shape functions, in this sense, automatically guarantees the internal compatibility) and at the interface with the surrounding elements.

~~Not Continuity~~

Nodal Continuity is enforced as part of

the assembly procedure. Attention should be paid in the formulation of kinematic elements (such as Euler-Bernoulli beam elements) for which the continuity involves the function along with the first derivative (the rotations). The continuity has to be guaranteed also at the frontier between adjacent edges of the elements. In this case a critical situation can be associated with the presence of curved edges.

2. the loads are applied in a consistent way (conformal)

This means that any kind of distributed external load is reported at the nodes by projecting it over the shape functions of the element. In other words, the external work should be written as:

$$\delta W_e = \int_V \underline{\underline{s}}_u^T \underline{\underline{N}}^T \underline{\underline{f}} dV = \underline{\underline{s}}_u^T \underbrace{\int_V \underline{\underline{N}}^T \underline{\underline{f}} dV}_{\text{consistent external loads}}$$

3. Numerical integration is carried out exactly

The integrals, entering the expression of $\underline{\underline{f}}$ and $\underline{\underline{k}}$, should be evaluated exactly. This operation is possible, in general, only if the element is not distorted.

Whenever the three previous requirements are not verified, it may happen that the total potential energy (and the strain energy) behaves differently from what expected.

In this sense, the model behaviour will be problem-dependent and inherently associated with the kind and degree of violation of the mentioned requirements. For instance, a model with a few distorted elements does not fulfill the third requirement. However, if the distorted elements are few, it is very likely that they will not affect the global convergence behaviour of the entire structural model.