



UNIVERSITÉ LIBRE DE BRUXELLES

SUMMARY

---

**Structural Analysis & Finite Elements  
MECA-H421**

---

*Author:*  
Enes ULUSOY

*Professors:*  
Lincy PYL  
Peter BERKE

Year 2016 - 2017

# Appel à contribution

## Synthèse Open Source



Ce document est grandement inspiré de l'excellent cours donné par Lincy PYL et Peter BERKE à l'EPB (École Polytechnique de Bruxelles), faculté de l'ULB (Université Libre de Bruxelles). Il est écrit par les auteurs susnommés avec l'aide de tous les autres étudiants et votre aide est la bienvenue ! En effet, il y a toujours moyen de l'améliorer surtout que si le cours change, la synthèse doit être changée en conséquence. On peut retrouver le code source à l'adresse suivante

<https://github.com/nenglebert/Syntheses>

Pour contribuer à cette synthèse, il vous suffira de créer un compte sur *Github.com*. De légères modifications (petites coquilles, orthographe, ...) peuvent directement être faites sur le site ! Vous avez vu une petite faute ? Si oui, la corriger de cette façon ne prendra que quelques secondes, une bonne raison de le faire !

Pour de plus longues modifications, il est intéressant de disposer des fichiers : il vous faudra pour cela installer  $\text{\LaTeX}$ , mais aussi *git*. Si cela pose problème, nous sommes évidemment ouverts à des contributeurs envoyant leur changement par mail ou n'importe quel autre moyen.

Le lien donné ci-dessus contient aussi un README contenant de plus amples informations, vous êtes invités à le lire si vous voulez faire avancer ce projet !

## Licence Creative Commons

Le contenu de ce document est sous la licence Creative Commons : *Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)*. Celle-ci vous autorise à l'exploiter pleinement, compte-tenu de trois choses :



1. *Attribution* ; si vous utilisez/modifiez ce document vous devez signaler le(s) nom(s) de(s) auteur(s).
2. *Non Commercial* ; interdiction de tirer un profit commercial de l'œuvre sans autorisation de l'auteur
3. *Share alike* ; partage de l'œuvre, avec obligation de rediffuser selon la même licence ou une licence similaire

Si vous voulez en savoir plus sur cette licence :

<http://creativecommons.org/licenses/by-nc-sa/4.0/>

Merci !

# Contents

<b>2 Models</b>	<b>1</b>
2.1 Successive modeling phases . . . . .	1
<b>3 Solid mechanics</b>	<b>2</b>
3.1 Continuum mechanics . . . . .	2
3.1.1 Statics . . . . .	2
3.1.2 Kinematics . . . . .	4
3.2 Linear elasticity . . . . .	5
3.2.1 Material law . . . . .	5
3.2.2 Strain energy . . . . .	6
<b>4 Formulations in linear elasticity</b>	<b>7</b>
4.1 Strong formulation . . . . .	7
4.1.1 Boundary conditions . . . . .	7
4.1.2 Mathematical properties of the governing equations . . . . .	7
4.1.3 Linear elasticity governed by elliptic PDE's . . . . .	8
4.2 Energy principles . . . . .	9
4.2.1 Principle of virtual work . . . . .	9
4.2.2 Variational formulations . . . . .	10
<b>5 Approximations</b>	<b>12</b>
5.1 Global vs. local approximation . . . . .	12
5.1.1 Global approximation . . . . .	12
5.1.2 Local approximation . . . . .	14
<b>6 Isoparametric elements</b>	<b>16</b>
6.1 Partition of the domain . . . . .	16
6.2 Classical elements . . . . .	16
6.3 Reference element and real elements . . . . .	17
6.4 Approximation on the reference element . . . . .	19
6.5 Construction of the shape function . . . . .	19
<b>7 Discretization by finite elements</b>	<b>21</b>
7.1 Weighted residual method . . . . .	21
7.1.1 Collocations by points . . . . .	21
7.1.2 Collocation by sub-domains . . . . .	21
7.1.3 Least square method . . . . .	22
7.1.4 Galerkin method . . . . .	22
7.2 Ritz analysis methods . . . . .	23
7.3 Properties of the finite element solution . . . . .	24

<b>8 2D elements in plane stress and plane strain</b>	<b>25</b>
8.1 Triangular elements . . . . .	25
8.1.1 TRIM-3 . . . . .	25
8.1.2 TRIM-6 . . . . .	27
8.2 Quadrangular elements . . . . .	28
8.2.1 REM-4 . . . . .	28
8.2.2 REM-8 and REM-9 . . . . .	28
8.3 Connecting elements of different types . . . . .	29
<b>9 Assembly and boundary conditions</b>	<b>30</b>
9.1 Assembly . . . . .	30
9.2 Boundary conditions . . . . .	30
9.2.1 Direct method . . . . .	31
9.2.2 Penalty method . . . . .	31
<b>10 Numerical integration</b>	<b>33</b>
10.1 Exploiting isoparametric properties . . . . .	33
10.2 Numerical integration . . . . .	33
10.3 Selection of the quadrature order . . . . .	34
10.3.1 Matrix singularity due to numerical integration . . . . .	34
10.3.2 Detection of skewed elements . . . . .	35
10.3.3 Full vs. reduced integration . . . . .	35
<b>11 System resolution</b>	<b>37</b>
11.1 Properties of the stiffness matrix . . . . .	37
11.2 Algorithms for solving linear systems . . . . .	37
11.2.1 Direct methods . . . . .	37
11.2.2 Cholesky decomposition . . . . .	38
11.2.3 Frontal method . . . . .	39
11.2.4 Iterative methods . . . . .	39
<b>12 Symmetry, repeatability, modularity</b>	<b>41</b>
12.1 Notion of symmetry . . . . .	41
12.2 Notion of repeatability and modularity . . . . .	41
<b>13 Verification &amp; Validation</b>	<b>43</b>
13.1 Criteria for convergence . . . . .	43
13.2 The patch test for solid elements . . . . .	44
13.3 A short glimpse into error estimation techniques . . . . .	44

# Chapter 2

## Models

### 2.1 Successive modeling phases

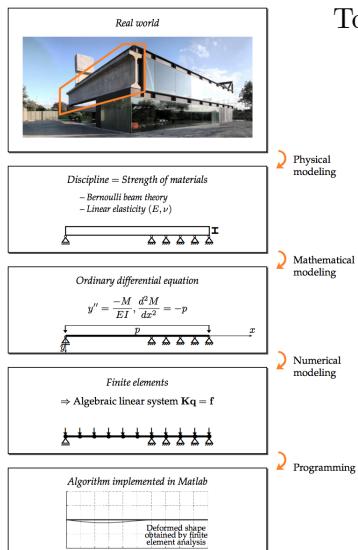


Figure 2.1

To implement a model on the computer we need:

- a **physical model**: step of defining the disciplines involved and (e.g. fluid dynamics) and the hypothesis regarding the material law (e.g. elasticity); In our example we can see a building where there is a huge beam. We want to model this and first this is a volume. We will represent it in 2D and the first question is "do I have shear stress". Yes there is a wall below leading to the small triangles on the figure, assuming that there is no displacement in y axis. Then the second question is "do I care of elastic behavior of concrete or is linear elasticity enough?" → assumption.
- a **mathematical model**: translation of the physical principles into mathematical language; In the example we have the relation with  $y''$  and the one concerning the load, which is the roof part. We are assuming here that the weight of the roof is equally distributed.

- a **numerical model**: implementing an algorithm able to solve the previous point equations; What we do is in fact cutting our element in several elements, separated by nodes. The distributed forces will then be applied on that nodes. **THE** equation for finite element is:  $Kq = f$  where  $K$  is a  $6 \times 6$  matrix and represents **stiffness**,  $q$  **displacement** and  $f$  **force** are  $6 \times 1$  vectors.
- a **computer model**: implementation of an in-house code or a commercial software product, based on the previous point.

In this course we will be using a displacement based finite element model, the only unknown is the displacement, then we can find the stresses.

Be aware that some steps of the process introduce errors. Indeed, the choice of the physical model, then the mathematical model (choice), the discretization (we solve for the nodes and not the whole model) and the computer-based model (inversion of matrix thousand and thousand times) are not perfect.

# Chapter 3

## Solid mechanics

### 3.1 Continuum mechanics

#### 3.1.1 Statics

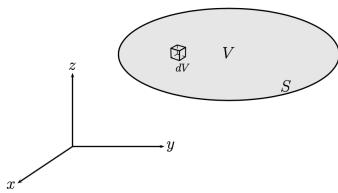


Figure 3.1

We will establish the governing equations in **continuum** mechanics. Consider a volume  $V$  delimited by a surface  $S$ , defined in a right-handed Cartesian coordinate. Continuum means, despite the microscopic description, that the material is assumed to behave as a continuum whole body. We can define the physical quantity in every point  $(x,y,z)$  of  $V$  by means of continuous function:

$$\rho = \lim_{dV \rightarrow 0} \frac{dm}{dV}. \quad (3.1)$$

In addition, we assume the **differentiability** that allows to write these equations in function of infinitesimal quantities. We also assume that the material is **homogeneous** and **isotropic** (same mechanical properties in all directions). Birth and propagation of cracks are causes of loss of continuity. In this case the continuity approach is not valid anymore. In numerical methods, it requires extensions as X-FEM and others.

There are two types of external forces:

- **Body forces  $\mathbf{b}$ :** acting throughout the volume  $V$ . This depending on position, the resultant:

$$\mathbf{f}^v = \int_V \mathbf{b}(x, y, z) \rho(x, y, z) dV. \quad (3.2)$$

In statics, gravity loads are the main body forces:

$$\mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}, \quad (3.3)$$

$z$  assumed to be the vertical axis oriented upwards.

- **Contact forces  $\mathbf{t}$ :** present at the contact between two points or surfaces. In practice it is either external forces on  $S$  or reactions at attachment points.

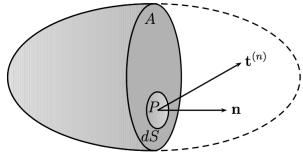


Figure 3.2

To introduce the notion of **Cauchy stresses**, let's cut a section  $A$  in the a volume  $V$ . The arbitrary surface  $dS$  on  $A$  is characterized by a resultant force  $d\mathbf{f}$  and a resultant moment  $d\mathbf{m}$ . The **Cauchy stress vector** is defined as:

$$\mathbf{t}^{(n)} = \lim_{dS \rightarrow 0} \frac{d\mathbf{f}}{dS}. \quad (3.4)$$

For the rest of the course we will assume  $d\mathbf{m}/dS = 0$ . Remark that  $\mathbf{t}^{(n)}$  is associated to a certain normal, if we make another cut  $A'$ , we will have another normal  $\mathbf{n}'$  and a different stress vector. We only have that  $\mathbf{t}^{(n)} = -\mathbf{t}^{(-n)}$  (action-reaction).

Since each direction is associated to a stress vector, we define the second-order **stress tensor**  $\bar{\tau}$ . For this, we make the stress vectors defines for the three coordinate planes related to the unit normals  $\mathbf{e}^{(1)}, \mathbf{e}^{(2)}, \mathbf{e}^{(3)}$  (see Figure 3.3). From this tensor, we can find any stress vector by projecting the tensor on the normal  $\mathbf{n}$  associated to the cutting plane:

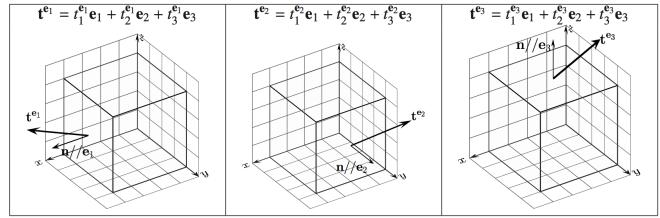
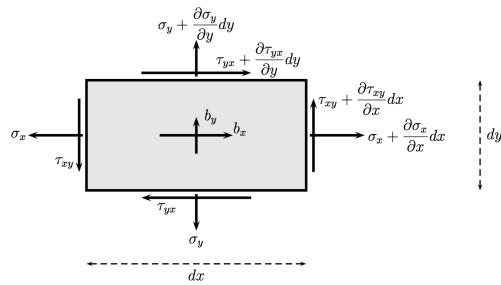


Figure 3.3

$$\bar{\tau} \cdot \mathbf{n} = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix}. \quad (3.5)$$

As reminder of the notation,  $\tau_{xx} = \sigma_x$  (normal stress,  $0 \leftrightarrow$  tension,  $< 0 \rightarrow$  compression) points the normal component in x direction,  $xy$  (shear stress) will points to the vector of the plane  $\perp x$  oriented to y and  $xz$  is the same but oriented to z. Stresses are measured in  $N/m^2 = Pa$ .

We have to make the difference between 0-order, 1-order and 2-order tensors. The first means that the same scalar value is associated to each direction of the 3D space (does not depend on the orientation). For a 1-order tensor  $\mathbf{v}$ , assuming a given orientation  $\mathbf{d}$ , a scalar value is associated to  $\mathbf{d}$  by  $v_d = \mathbf{d} \cdot \mathbf{v}$ . This scalar change in function of the considered direction. And for the last, we have a different vector for any direction.



It is finally time to derive the equilibrium equations. Instead of computing this for every point of a body, we will use an infinitesimal element  $dx dy$  in 2D. By isolating this element, all forces (body and surface) acting on it should be balanced. The balance on the x-axis gives:

Figure 3.4

$$\begin{aligned} b_x dx dy + \left( \frac{D\sigma_x}{\partial x} dx \right) dy + \left( \tau_{yx} + \frac{\partial \tau_{yx}}{\partial y} dy \right) dx - \sigma_x dy - \tau_{yx} dx &= 0 \\ \Leftrightarrow b_x dx dy + \frac{\partial \sigma_x}{\partial x} dx dy + \frac{\partial \tau_{yx}}{\partial y} dx dy &= 0 \\ \Leftrightarrow b_x + \frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} &= 0 \end{aligned} \quad (3.6)$$

By using indicial notation, we can generalize this approach in 3D and get a set of three equilibrium equations in translation:

$$b_i + \tau_{ji,j} = 0 \quad (3.7)$$

where  $i$  is a free or unrepeated index and  $j$  a summed or dummy index. Here are some rules:

- $\delta_{ij}$  is the **Kronecker delta** and  $= 1$  if  $i = j$ ,  $= 0$  otherwise;
- $\epsilon_{ijk}$  is the permutation symbol which is  $= 1$  if  $ijk$  makes a positive permutation,  $= -1$  if negative permutation and  $= 0$  otherwise (see syllabus if don't remember);
- $u_i v_i$  is a scalar product of  $\mathbf{u}$  and  $\mathbf{v}$ ;
- $\epsilon_{ijk} u_j v_k$  is a cross product;
- $\epsilon_{ijk} \partial_j u_k$  is the curl of  $\mathbf{u}$  ( $\nabla \times \mathbf{u}$ );
- Gauss theorem in indicial notation:

$$\int_V u_{i,i} dV = \oint u_i n_i dS. \quad (3.8)$$

Now we have also to verify the rotation equilibrium. If the reference point is the origin of the axes, we denote  $x_i$  the current position, then the rotation equilibrium for an arbitrary volume  $V' \in V$  delimited by  $S'$  is:

$$\int_{V'} \epsilon_{ijk} x_j b_k dV' + \oint_{S'} \epsilon_{ijk} x_j t_k^{(n)} dS' = 0 \quad (3.9)$$

where we applied the definition of the moment position  $\times$  force. We know that  $t_k^{(n)} = \tau_q k n_q$ . By using this and the Gauss theorem we obtain:

$$\begin{aligned} \int_V \epsilon_{ijk} [x_j b_k + (x_j \tau_{qk}, q)] dV' &= \int_V \epsilon_{ijk} [x_j b_k + x_j \tau_{qk,q} + x_{j,q} \tau_{qk}] dV' \\ &= \int_V \epsilon_{ijk} \left[ x_j \underbrace{(b_k + \tau_{qk,q})}_{=0} + x_{j,q} \tau_{qk} \right] dV' = \int_V \epsilon_{ijk} x_{j,q} \tau_{qk} dV' = 0. \end{aligned} \quad (3.10)$$

Remark that  $x_{j,q} = \delta_{jq}$ , and since  $V' \in V$  is completely arbitrary the integral must vanish:

$$\epsilon_{ijk} \tau_{jk} = 0. \quad (3.11)$$

We can conclude that **in the absence of concentrated body moments, the stress tensor is symmetric**. We revise our equation to

$\tau_k^{(n)} = \tau_{kq} n_q \quad ; \quad b_i + \tau_{ij,j} = 0 \quad ; \quad \tau_{ij} = \tau_{ji}$

(3.12)

### 3.1.2 Kinematics

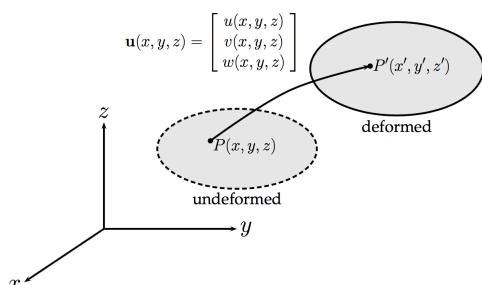


Figure 3.5

Continuum mechanics is also concerned by the way the volume is deformed through displacement and strains. Consider the volume  $V$  which is deformed in  $V'$  by applying a **displacement** on each point of  $V$ . In much applications the displacement can be assumed to be much smaller than the dimensions of the volume, leading to the **infinitesimal strain theory**, also called **small displacement-gradient theory**.

This simplifies our live because we assume the deformed volume to remain as the initial one and we can perform the integrals interchangeably on the initial or deformed configuration. This is valid for stiff materials like steel. Other flexible materials are the scope of non linear mechanics. The **linear strain tensor** is derived from the displacement field:

$$\epsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}) \quad \bar{\boldsymbol{\epsilon}} = \begin{bmatrix} \epsilon_x & \frac{\gamma_{xy}}{2} & \frac{\gamma_{xz}}{2} \\ \frac{\gamma_{yx}}{2} & \epsilon_y & \frac{\gamma_{yz}}{2} \\ \frac{\gamma_{zx}}{2} & \frac{\gamma_{zy}}{2} & \epsilon_z \end{bmatrix} \quad (3.13)$$

where the  $\epsilon_i$  are the axial strain and the  $\gamma_{ij}$  are the shear strain. **Don't forget the importance of small displacements!**

## 3.2 Linear elasticity

### 3.2.1 Material law

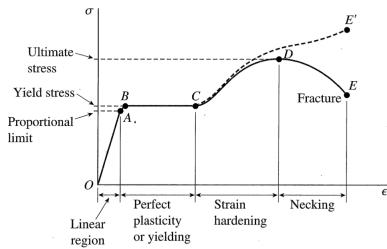


Figure 3.6

The understanding of the mechanical response requires a link between stresses and strains. This is done by the **material law**. It is find experimentally, using a tensile test machine that loads the material in tension at a constant speed until fracture. The **nominal axial stress** and the **average axial strain** are obtained respectively by dividing the force and the displacement by the initial surface and length of the sample:

$$\sigma = \frac{P}{A_0} \quad \epsilon = \frac{\delta}{L_0} \quad (3.14)$$

On Figure 3.6, the O state corresponds to no strain no stress, then a straight line to A. The slope of OA is called the **modulus of elasticity** or the **Young's modulus**, noted E [ $N/m^2$ ]. After A, the relation is no longer linear. In AB the strain increases more rapidly than the stress, until a plateau BC where large strains are obtain without increase of load, **perfect plasticity**. The constant stress at this stage is the **yield stress**.

After this, the material **strain harden**, it resists to further deformation. The maximum stress is the **ultimate stress** on D. Then, the section A is shrunk and the bar is necking. The load decreases until the failure. By using the "true" cross-section area  $A_{true}$ , we can draw a true stress-strain curve CE'. We will focus on **linear elastic materials**. Elasticity points that  $\sigma$  is a unique function of  $\delta$  and that the material recovers initial state when unloaded. Linearity points to the proportionality.

Additionally to the axial deformation, in prismatic bar is accompanied by a **lateral contraction** from the very beginning of the load. In the elastic domain, this is the **Poisson effect**:

$$\epsilon_x = \frac{\sigma_x}{E}, \quad \epsilon_y = \epsilon_z = -\frac{\nu \sigma_x}{E} \quad (3.15)$$

where  $0 \leq \nu \leq 0.5$  is the Poisson coefficient. Let's remark that similarly to the Hooke's law, a shear version can be:

$$\tau = G\gamma \quad G = \frac{E}{2(1+\nu)} \quad (3.16)$$

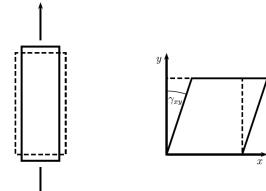


Figure 3.7

where the shear strain  $\gamma_{xy}$  can be interpreted as the angular deformation in xy. In 3D, as  $\tau_{ij}$  and  $\epsilon_{ij}$  are second order tensors, we need a fourth-order tensor  $C_{ijkl}$  such that  $\tau_{ij} = C_{ijkl}\epsilon_{kl}$ . This is simplified for **isentropic elastic materials**:

$$\tau_{ij} = \lambda\epsilon_{kk} + 2\mu\epsilon_{ij} \quad (3.17)$$

where  $\lambda, \mu$  are the **Lamé constants**, defined as:

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)} = G. \quad (3.18)$$

Finally, if we replace we get

### Stress-strain relationship in isotropic elasticity

$$\epsilon_{ij} = \frac{1}{E}[(1+\nu)\tau_{ij} - \nu\delta_{ij}\tau_{kk}]. \quad (3.19)$$

In general material have also a non-linear behavior, but as in engineering we design the material to remain below the linear limit, we can make the approx.

### 3.2.2 Strain energy

When a load is applied, the external work of the force is converted into strain energy. Indeed, the application of  $\sigma_x$  induces an extension  $\epsilon_x dx$ , physically the energy for a volume is:

$$dW = \frac{1}{2}\sigma_x\epsilon_x dx dy dz. \quad (3.20)$$

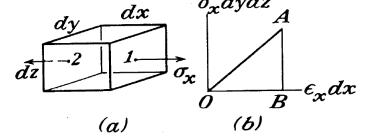


Figure 3.8

In the general case, the conservation of energy for linear elastic materials, the work cannot depend on the order of magnitude of the load. In other word, we have to multiply stress and strain tensors component by component:

$$\begin{aligned} dW &= W_V dx dy dz = \frac{1}{2}\tau_{ij}\epsilon_{ij} dx dy dz. \\ &= \frac{1}{2}(\sigma_x\epsilon_x + \sigma_y\epsilon_y + \sigma_z\epsilon_z + \tau_{xy}\epsilon_{xy} + \tau_{xz}\epsilon_{xz} + \tau_{yz}\epsilon_{yz} + \tau_{yx}\epsilon_{yx} + \tau_{zx}\epsilon_{zx} + \tau_{zy}\epsilon_{zy}) dx dy dz \\ &= \frac{1}{2}(\sigma_x\epsilon_x + \sigma_y\epsilon_y + \sigma_z\epsilon_z + \tau_{xy}\gamma_{xy} + \tau_{xz}\gamma_{xz} + \tau_{yz}\gamma_{yz}) dx dy dz \end{aligned} \quad (3.21)$$

where  $W_V$  is the **strain energy density**:  $W_V = \int_{\epsilon_{ij}} \tau_{ij} d\epsilon_{ij}$ .

We see the utility of defining  $\epsilon_{xy} = \frac{1}{2}\gamma_{xy}$  in the equation. Using the Hooke's law:

$$\begin{aligned} W_V &= \frac{1}{2E}(\sigma_x^2 + \sigma_y^2 + \sigma_z^2) - \frac{\nu}{E}(\sigma_x\sigma_y + \sigma_x\sigma_z + \sigma_y\sigma_z) + \frac{1}{2G}(\tau_{xy}^2 + \tau_{xz}^2 + \tau_{yz}^2) \\ &= \frac{1}{2}\lambda(\epsilon_x^2 + \epsilon_y^2 + \epsilon_z^2) + G(\epsilon_x^2 + \epsilon_y^2 + \epsilon_z^2) + \frac{1}{2}G(\gamma_{xy}^2 + \gamma_{xz}^2 + \gamma_{yz}^2). \end{aligned} \quad (3.22)$$

We see that  $W_V$  is always **positive**. Let's define the dual  $W_V^* = \int_{\tau_{ij}} \epsilon_{ij} d\tau_{ij}$ , such that:

$$W_V + W_V^* = \tau_{ij}\epsilon_{ij}. \quad (3.23)$$

The strain energy  $W$  and the complementary are obtained by  $\int_V W_V dV$ , with  $W_V = W_V^* = \frac{1}{2}\tau_{ij}\epsilon_{ij}$ .

# Chapter 4

## Formulations in linear elasticity

### 4.1 Strong formulation

The equilibrium equations omitting concentrated body moments and the material law in linear elasticity for isotropic and homogeneous materials are:

$$b_i + \tau_{ij,j} = 0, \quad \epsilon_{ij} = \frac{1}{E}[(1+\nu)\tau_{ij} - \nu\delta_{ij}\tau_{kk}]. \quad (4.1)$$

Since there is a linear relation between stresses and strains, strain tensor  $\epsilon_{ij}$  derives itself from the displacement field, we can rewrite the equilibrium equations in function of the displacements. This leads to the **displacement-based finite element method**, the unknowns are the displacements. All the set of equations are encompassed under the **strong formulation** therminology, because they require to be satisfied **locally**, at each point of the domain  $V$ .

#### 4.1.1 Boundary conditions

The main classes of boundary conditions are:

- **essential boundary conditions** (Dirichlet): the displacement is prescribed on a portion of the external surface  $S$ , called  $S_u$ ;
- **natural boundary conditions** (Neumann): the contact force is imposed on another portion of the external surface  $S$ , called  $S_t$ .

#### 4.1.2 Mathematical properties of the governing equations

First, the **superposition principle** is respected by stresses, strains and displacements because linearity of the static and kinematic equations in linear elasticity (assuming small displacements/strains). Indeed if  $\tau_{ij}^A$  and  $\tau_{ij}^B$  are stress tensors associated to load cases A and B,  $\tau_{ij}^A + \tau_{ij}^B$  is the solution for the load case A + B.

Secondly, for given surface and body forces, the uniqueness of the solution for the governing equations is guaranteed.

*Proof.* As a counterargument, let's assume that there exist two solution  $u_i^{(1)}$  and  $u_i^{(2)}$  and their difference  $u'_i = u_i^{(1)} - u_i^{(2)}$ . Let's do the same for strains and stresses:

$$\epsilon'_{ij} = \epsilon_{ij}^{(1)} - \epsilon_{ij}^{(2)}, \quad \tau'_{ij} = \tau_{ij}^{(1)} - \tau_{ij}^{(2)}. \quad (4.2)$$

Since the body forces are external and identical for the two solutions, we get from the equilibrium equations:

$$\epsilon_{ij,j}^{(1)} - \epsilon_{ij,j}^{(2)} = \epsilon'_{ij,j} = 0. \quad (4.3)$$

On the other hand, the strain energy of the difference is:

$$W' = \frac{1}{2} \int_V \epsilon'_{ij} \tau'_{ij} dV = \frac{1}{2} \int_V \frac{1}{2} (u'_{i,j} + u'_{j,i}) \tau'_{ij} dV = \frac{1}{2} \int_V u'_{i,j} \tau'_{ij} dV, \quad (4.4)$$

by symmetry and as i, j are repeated. If we use integration by part and Gauss theorem, we get:

$$\begin{aligned} W' &= \frac{1}{2} \int_V (u'_i \tau'_{ij})_{,j} dV - \frac{1}{2} \int_V u'_i \tau'_{ij,j} dV \\ &= \frac{1}{2} \oint_S u'_i \tau'_{ij} n_j dV = \frac{1}{2} \oint_S u'_i \bar{\tau}'_i^{(n)} dV \end{aligned} \quad (4.5)$$

To guarantee the uniqueness of the solution, so to have  $W' = 0$ , the boundary conditions must satisfy:

1. the displacement field  $u_i$  is prescribed on the entire surface, leading directly to  $u'_i = 0$
2. the contact forces  $t_i^{(n)}$  are imposed over the entire surface while overall equilibrium is satisfied, making  $t_i^{(n)} = 0$
3. displacements  $u_i$  are prescribed on one part of the surface ( $u_i = \bar{u}_i$  on  $S_u$ ), while contact forces  $\bar{\tau}_i^{(n)}$  are imposed on the rest of the surface ( $\bar{\tau}_i^{(n)} = \bar{\tau}_i^{(n)}$  on  $S_t$ ), so that either  $u'_i = 0$  or  $\bar{\tau}_i^{(n)} = 0$ . The  $\bar{\cdot}$  symbol indicates that the values are prescribed.

□

This explains why it is not allowed to impose both the displacement and the force on the same point and the same direction. Example on Figure 4.1.

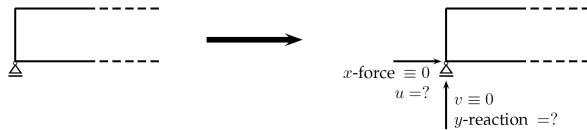


Figure 4.1

#### 4.1.3 Linear elasticity governed by elliptic PDE's

A wide range of systems are governed by partially differential equations. It can be proven that the governing equations for linear elasticity constitute a set of **elliptic second-order partial differential equations**. An implication of this kind of system is that a small disturbance at one point P in the domain V has an impact on the whole body, and inversely a point P is influenced by the entire boundary surface S. This is in contrast with other types of PDE's (parabolic and hyperbolic).

## 4.2 Energy principles

### 4.2.1 Principle of virtual work

The difficulty of the equilibrium equations is that they must be satisfied in each point of the domain. Approches based on the energy principles have been proposed. The equilibrium equations were:

$$b_i + \tau_{ij,j} = 0. \quad (4.6)$$

As this has to be satisfied everywhere, we can multiply by a virtual displacement  $\hat{u}_i$  and integrate over the whole domain  $V$ :

$$\int_V (b_i + \tau_{ij,j}) \hat{u}_i dV = 0 \quad \forall \hat{u}_i. \quad (4.7)$$

If we use Gauss theorem and decompose the displacement in its symmetric and anti-symmetric part we get:

$$\begin{aligned} \int_V \tau_{ij,j} \hat{u}_i dV &= \int_V (\tau_{ij} \hat{u}_i)_{,j} dV - \int_V \tau_{ij} \hat{u}_{i,j} dV = \oint_S \tau_{ij} \hat{u}_i n_j dS - \int_V \tau_{ij} \hat{u}_{i,j} dV \\ &= \oint_S \tau_i^{(n)} \hat{u}_i dS - \int_V \tau_{ij} \frac{1}{2} (\hat{u}_{i,j} + \hat{u}_{j,i}) dV - \int_V \tau_{ij} \frac{1}{2} (\hat{u}_{i,j} - \hat{u}_{j,i}) dV \\ &= \oint_S \tau_i^{(n)} \hat{u}_i dS - \int_V \tau_{ij} \hat{\epsilon}_{ij} dV \end{aligned} \quad (4.8)$$

where the last integral vanishes as we have the multiplication of a symmetric tensor with an anti-symmetric one. In this way we can express the

#### Total virtual work principle

$$\int_V b_i \hat{u}_i dV + \oint_S \tau_i^{(n)} \hat{u}_i dS - \int_V \tau_{ij} \hat{\epsilon}_{ij} dV = 0 \quad \forall \hat{u}_i. \quad (4.9)$$

Remark that **the forces are real and the displacement are virtual**. This principle states that:

- if the body is in equilibrium, then the total virtual work is equal to zero for any virtual displacement (direct version);
- if the total virtual work is equal to zero for any virtual displacement, then the body is in equilibrium (reciprocal version).

We can redefine the spaces of the displacement functions by introducing two concepts:

- $u_i$  is **kinematically admissible** if it satisfies the geometric boundary conditions, i.e. if  $u_i = \bar{u}_i$  on the portion  $S_u$  of the external surface  $S$ .
- $u_i$  is **kinematically homogeneous** if it vanishes on the boundary conditions, i.e. if  $u_i = 0$  on the portion  $S_u$  of the external surface  $S$ .

If the choice of the virtual displacement is restricted to **kinematically homogeneous** fields, then the simplification:

$$\oint_S \tau_i^{(n)} \hat{u}_i dV = \oint_{S_t} \bar{\tau}_i^{(n)} \hat{u}_i dS. \quad (4.10)$$

This simplification is motivated by the fact that we don't need all the unknown reactions on the boundary conditions. This integral is referred to as the **weak integral formulation**. Since the

fundamental variable is the real displacement field  $u_i$ , the previous theorem can be simplified as:

**Virtual displacement theorem for kinematically homogeneous virtual displacement**

Let  $\bar{\tau}_i^{(n)}, b_i$  be a system of external forces acting on a body  $V$ . Among all kinematically admissible displacement fields ( $\mathbf{u} = \bar{\mathbf{u}}$ ),  $\mathbf{u}$  is the solution of the equilibrium problem if and only if:

$$\int_V b_i \hat{u}_i dV + \oint_{S_t} \tau_i^{(n)} \hat{u}_i dS - \int_V \tau_{ij}(\mathbf{u}) \hat{\epsilon}_{ij} dV = 0 \quad (4.11)$$

for all kinematically homogeneous virtual displacement fields  $\hat{u}_i$ .

Let's define the following terms:

$$\begin{aligned} a(\mathbf{u}, \hat{\mathbf{u}}) &= \int_V \tau_{ij}(\mathbf{u}) \hat{\epsilon}_{ij} dV & \varphi(\mathbf{u}) &= \int_V b_i \hat{u}_i dV + \oint_{S_t} \tau_i^{(n)} \hat{u}_i dS \\ &\Rightarrow a(\mathbf{u}, \hat{\mathbf{u}}) - \varphi(\mathbf{u}) = 0 \end{aligned} \quad (4.12)$$

where  $a(\mathbf{u}, \hat{\mathbf{u}})$  is bilinear. It is important to remark that we only deal with first order derivatives while we have second order in the strong form. This means that besides computing point per point we are interested in integral quantities over the whole domain.

#### 4.2.2 Variational formulations

We can rewrite (4.11) by considering the virtual displacements as small arbitrary variations  $\delta u_i$  of the real displacements (and associated strain tensor variations  $\delta \epsilon_{ij}$ ):

$$\int_V b_i \delta u_i dV + \oint_{S_t} \tau_i^{(n)} \delta u_i dS - \int_V \tau_{ij} \delta \epsilon_{ij} dV = 0 \quad \forall \delta u_i | \delta u_i = 0 \text{ on } S_u. \quad (4.13)$$

Let's introduce the **potential energy of the external loads  $\mathbf{U}$**  and the **strain energy  $\mathbf{W}$** :

$$\begin{aligned} \int_V b_i \delta u_i dV + \oint_{S_t} \tau_i^{(n)} \delta u_i dS &= \delta \left( \int_V b_i u_i dV + \oint_{S_t} \tau_i^{(n)} u_i dS \right) = -\delta U \\ \int_V \tau_{ij} \delta \epsilon_{ij} dV &= \delta W. \end{aligned} \quad (4.14)$$

Indeed, notice that the strain energy can be expressed by means of the bilinear form:

$$W(\mathbf{u}) = \frac{1}{2} a(\mathbf{u}, \mathbf{u}) \quad \Rightarrow \delta W(\mathbf{u}) = \frac{1}{2} a(\delta \mathbf{u}, \mathbf{u}) + \frac{1}{2} a(\mathbf{u}, \delta \mathbf{u}) + \frac{1}{2} a(\delta \mathbf{u}, \delta \mathbf{u}). \quad (4.15)$$

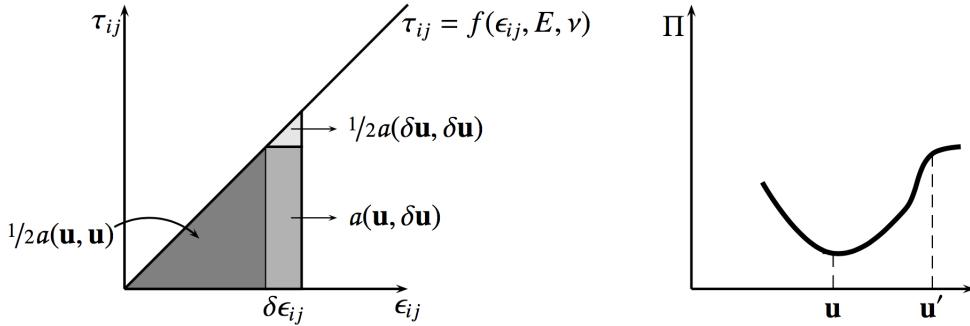


Figure 4.2

This is represented on Figure 4.2 left where  $\delta W(\mathbf{u}) \approx a(\mathbf{u}, \delta\mathbf{u})$  since  $a$  is symmetric and  $\frac{1}{2}a(\delta\mathbf{u}, \mathbf{u})$  is a second order that can be neglected. We have:

**Variational formulation of the virtual displacement theorem**

$$\delta\Pi = \delta(a(\mathbf{u}, \mathbf{u}) - \varphi(\mathbf{u})) = \delta(U + W) = 0 \text{ at equilibrium,} \quad (4.16)$$

where  $\Pi$  is the **total potential energy**.

The total potential energy is stationary for variations of admissible displacements. This is a **minimum** formulation:

*Proof.* Let's define another kinematically admissible displacement  $\mathbf{u}' = \mathbf{u} + \mathbf{v}$ , where  $\mathbf{v}$  is a kinematically homogeneous displacement field. The definition of the total potential energy gives:

$$\begin{aligned} \Pi(\mathbf{u}') &= \Pi(\mathbf{u} + \mathbf{v}) = \frac{1}{2}a(\mathbf{u} + \mathbf{v}, \mathbf{u} + \mathbf{v}) - \varphi(\mathbf{u} + \mathbf{v}) \\ &= \frac{1}{2}a(\mathbf{u}, \mathbf{u}) + \frac{1}{2}a(\mathbf{v}, \mathbf{v}) + a(\mathbf{u}, \mathbf{v}) - \varphi(\mathbf{u}) - \varphi(\mathbf{v}) \\ \Pi(\mathbf{u}) &= \frac{1}{2}a(\mathbf{u}, \mathbf{u}) - \varphi(\mathbf{u}) \\ \Rightarrow \Pi(\mathbf{u}') - \Pi(\mathbf{u}) &= \frac{1}{2}a(\mathbf{v}, \mathbf{v}) + \underline{a(\mathbf{u}, \mathbf{v})} - \underline{\varphi(\mathbf{v})} \end{aligned} \quad (4.17)$$

where the last two terms vanishes since  $\mathbf{v}$  is kinematically homogeneous. The only remaining term is always  $\geq 0$ , proving that the total potential energy is a minimum at the solution.  $\square$

# Chapter 5

## Approximations

### 5.1 Global vs. local approximation

#### 5.1.1 Global approximation

The first type of approximation consists in choosing a type of function (often polynomial) with parameters  $a_i$  and defining  $u_{approx}$  fitting the measurement points. Assume that we have the exact measurement  $u_{exact,i}$  at 3 different point and that we want to approximate that by a second order polynomial:

$$u_{exact}(x) \approx u_{approx}(x) = a_1 + a_2x + a_3x^2. \quad (5.1)$$

By imposing this function to equal the exact values at the different points we get a system of three equations:

$$\begin{cases} a_1 + a_2x_1 + a_3x_1^2 = u_{exact,1} \\ a_1 + a_2x_2 + a_3x_2^2 = u_{exact,2} \\ a_1 + a_2x_3 + a_3x_3^2 = u_{exact,3} \end{cases} \quad (5.2)$$

As we have as many parameters as the number of measurement points, we have a single solution for the system. The approximation is often presented in the form:

$$u_{approx}(x) = [p_1(x) \ p_2(x) \ \dots] \begin{bmatrix} a_1 \\ a_2 \\ \vdots \end{bmatrix} = \mathbf{p}(x)^T \mathbf{a}. \quad (5.3)$$

where  $\mathbf{p}(x)$  is a set of linearly independent functions, sometimes called the **basis** and  $\mathbf{a}$  are the parameters of the approximation. Non linear expansion of parameters can also be encountered in a more advanced regression. The  $a_i$  suffer from a lack of physical meaning, this is why we use the **nodal representation** where we explicitly use the measurement:

$$u_{approx} = u_{exact,1}N_1(x) + u_{exact,2}N_2(x) + \dots = \mathbf{N}(x)^T \mathbf{q} \quad (5.4)$$

where  $\mathbf{N}(x)$  contains the interpolation functions and  $\mathbf{q}$  is the N exact nodal values. A widespread nodal approximation is the Lagrange interpolation, expressed assuming  $x_i \neq x_j$ :

$$N_i(x) = \prod_{j=1, j \neq i}^N \frac{x - x_j}{x_i - x_j}. \quad (5.5)$$

For example the first equation of the system will be in the case of three measurements:

$$N_1(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)}. \quad (5.6)$$

The nodal approximations are defined to be interpolant ( $u_{approx}(x_i) = u_{exact}(x_i)$ ), leading to:

$$N_i(x_j) = \delta_{ij} \quad (5.7)$$

Schemes are global, since the parameter  $a_i$  (or  $u_i$ ) hold for the whole domain. This has a direct drawback when the number of samples increase. Indeed let's look to the figures below, at the beginning increasing the order of the polynomial allows a better fitting but then we get oscillations. This is known as **Runge's phenomenon**.



Figure 5.1

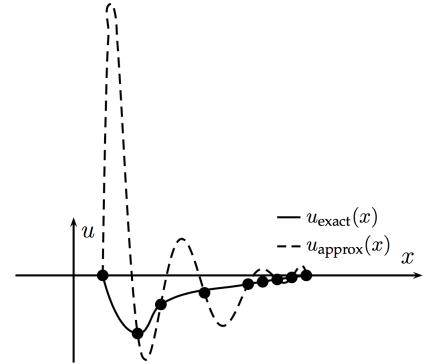


Figure 5.2

The problem can be mitigated by using a **least square approach**. Instead of a pure interpolation that forces to pass exactly on the samples, we will use an **ordinary least square** model where the order of the polynomial  $m$  will be independent of the number of sample points  $N$  ( $m \leq N$ ). The parameters are obtained by minimization of the error  $e_{LS}$  between the approx and the exact function:

$$\mathbf{a} = \arg \min_{\mathbf{a}} \left\{ e_{LS} \equiv \frac{1}{2} \sum_{i=1}^N (u_{exact,i} - \mathbf{p}(x_i)^T \mathbf{a})^2 \right\} = (\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T \mathbf{q}; \quad (5.8)$$

where  $\mathbf{P}$  contains the basis evaluated at the sample points  $x_i$ :

$$\mathbf{P} = \begin{bmatrix} p_1(x_1) & p_2(x_1) & \dots & p_{m+1}(x_1) \\ \vdots & & & \vdots \\ p_1(x_N) & p_2(x_N) & \dots & p_{m+1}(x_N) \end{bmatrix} \quad (5.9)$$

To give a higher influence on the samples located close to the point  $x$  where the prediction is required, an additional weight function can be used to obtain a higher impact on the nodes  $x_i$  close to  $x$ . This  $w_i = w(x; x_i)$  is usually a function decreasing monotonically with the distance  $\|x - x_i\|$ . This makes the minimum depends on  $x$ :

$$\begin{aligned} \mathbf{a}(x) &= \arg \min_{\mathbf{a}} \left\{ e_{LS} \equiv \frac{1}{2} \sum_{i=1}^N w_i (u_{exact,i} - \mathbf{p}(x_i)^T \mathbf{a})^2 \right\} = (\mathbf{P}^T \mathbf{W} \mathbf{P})^{-1} (\mathbf{P}^T \mathbf{W}) \mathbf{q}; \\ \mathbf{W} &= \begin{bmatrix} w_1 \equiv w(\|x - x_1\|) & 0 & \dots \\ 0 & w_2 \equiv w(\|x - x_2\|) & \dots \\ \vdots & & \vdots \end{bmatrix} \end{aligned} \quad (5.10)$$

If the approximation is non zero in a only in a vicinity around the guess point  $x$ , and vanishes everywhere else, the technique is called **moving least square**.

### 5.1.2 Local approximation

Here we divide from the beginning a local approximation through **nodal approximation by sub-domain**:

- domain  $V$  is divided into sub-domains  $V^e$ ;
- an approx is build for each  $V^e$ , the parameter can depend on the other sub-domains parameters (splines in CAD);
- the nodal approx on each  $V^e$  only involves the variables attached to the nodes within  $V^e$  and its boundary;
- functions  $u^e(x)$  are continuous within  $V^e$  and respect continuity conditions across sub-domains;
- $V^e$  are the **finite elements**
- the points where the approximations are forced to coincide with the exact value are the **interpolation nodes**; the coordinates of those points are the **nodal coordinates**;
- the values  $u_i = u_{approx}(x_i) \equiv u_{exact}(x_i)$  are the nodal variables.

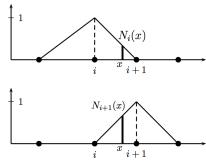


Figure 5.3

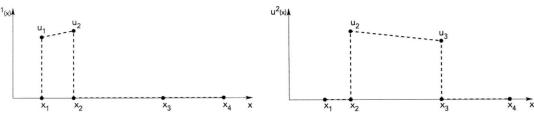
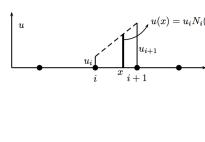


Figure 5.4

The simplest approximation consists in a piecewise first-order interpolation. On the above figures we can see an illustration in a one-dimensional problem with four nodes, on  $V^1$  only the first two nodes are taking place in the approx:

$$u_{approx}^{V^1} = N_1(x)u_1 + N_2u_2, \quad (5.11)$$

with the definition (5.5) and same for  $u_{approx}^{V^2}$  and  $u_{approx}^{V^3}$ .  $N_i$  are also called **shape functions**. This is illustrated on Figure 5.3. We see that the shape function is associated to a given node  $x_i$  and is different from zero on the elements containing it. This principle can be generalized in 3D (Figure 5.6).

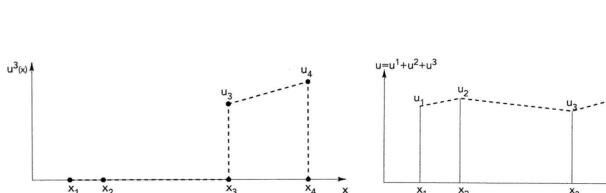


Figure 5.5

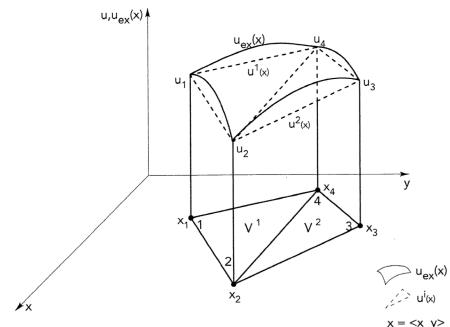


Figure 5.6

The finite element approximation offers 4 advantages:

- local behavior, one node influence only its neighborhood;
- low order polynomial in order to avoid the Runge's effect;
- interpolation at the nodes;
- physical interpretation of the coefficients.

Moreover as the shape function is non zero only for the  $x_i$  belonging to the element  $e$ , we can write for the whole discretized domain:

$$u_{approx} = N^e(x)q^e = N(x)q \quad (5.12)$$

where  $q^e$  contains the nodal values of  $u$  on the element, whereas  $q$  nodal values for the whole domain.

# Chapter 6

## Isoparametric elements

### 6.1 Partition of the domain

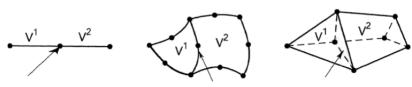


Figure 6.1

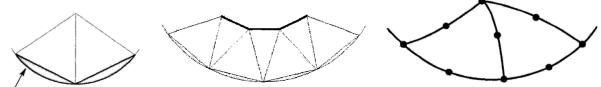


Figure 6.2

As we have seen, any domain  $V$  must undergo a sub-domain  $V^e$ . In dimension 1 the elements are separated by a point, 2 a curve and 3 a surface. The connection of all elements should be as closely as possible similar to the real domain. Discretization errors can occur for complex geometries. We can solve the issue by decreasing the size of the elements (mesh refreshment) or using elements with curved boundaries (Figure 6.2).

Nodes and elements must satisfy:

1. each element is defined by the coordinates of the nodes located in the element. In the majority of elements available in the literature, the nodes are on the boundary of the elements.
2. A portion of the boundary between two elements must be identically defined for both elements.

### 6.2 Classical elements

The order of the element is directly related to the number of nodes on its boundary and vice-versa. Most elements are first, second or third-order, enabling to model linear quadratic or cubic boundaries.

Dimension	Order	Linear	Quadratic	Cubic
1D		—●—●—●—	—●—●—●—●—●—	—●—●—●—●—●—●—●—●—
2D (triangular)		△	▲	▲
2D (quadrangular)		□	■	■
3D (tetrahedral)		■	■	■
3D (hexahedral)		■	■	■
3D (prismatic)		■	■	■

Figure 6.3

### 6.3 Reference element and real elements

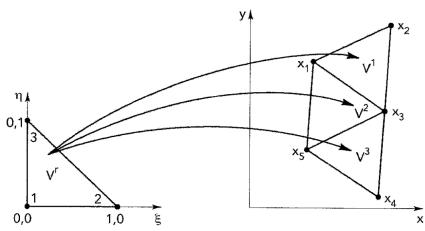


Figure 6.4

The finite element is the basic element on which the governing equations will be transposed. To avoid complex analytical operations, we will use a **reference element** on which a mapping of the **real element** (parent element) has to be performed. By this approach, the shape functions are written for the reference and then converted by linear transformation to the real. In 2D, the reference element is defined over a  $(\xi, \eta)$ -space, while the real elements are defined over the usual  $(x, y)$ -space.

To perform this operation, a transformation  $\mathcal{T}$  is required:

$$\mathcal{T} : \xi \rightarrow x\xi = \bar{N}^e(\xi)x^e \quad (6.1)$$

where  $x^e$  are the coordinates of the nodes of the element  $V^e$  and  $\bar{N}^e(\xi)$  are the **geometric transformation functions** which must follow the same rules as shape functions. This transformation allows to write the analytical definition for each  $V^e$  with respect to  $\xi$  on a simple element  $V^r$ . The function  $u(x)$  to be approximated can be written as  $u(\xi)$ .

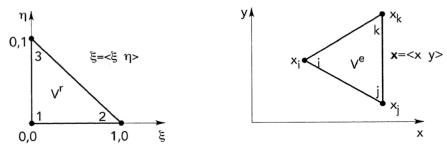


Figure 6.5

Let's make an example with a triangular element. On the figure is represented the reference and the parent element of a three-node triangular element. The reference is defined as:

$$\xi \geq 0, \quad \eta \geq 0, \quad \xi + \eta \leq 1. \quad (6.2)$$

Let us consider a real element defined by three nodes  $i, j, k$  and the transformation  $\mathcal{T}$  such that:

$$x(\xi, \eta) = [1 - \xi - \eta \quad \xi \quad \eta] \begin{bmatrix} x_i \\ x_j \\ x_k \end{bmatrix}, \quad y(\xi, \eta) = [1 - \xi - \eta \quad \xi \quad \eta] \begin{bmatrix} y_i \\ y_j \\ y_k \end{bmatrix}. \quad (6.3)$$

$\mathcal{T}$  verifies the following properties:

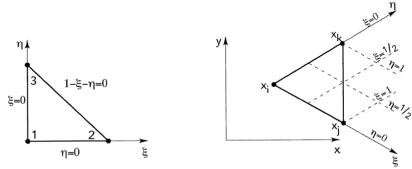
1. the nodes  $(0,0), (0,1)$  and  $(1, 0)$  from  $V^r$  are converted into  $(x_i, y_i), (x_j, y_j)$  and  $(x_k, y_k)$ :
2. each boundary of  $V^r$  is transformed into a boundary of  $V^e$ . For example, the line between  $(1, 0)$  and  $(0, 1)$  is transformed into  $1 - \xi - \eta = 0$  or  $\eta = 1 - \xi$ :

$$x = [0 \quad \xi \quad 1 - \xi] \begin{bmatrix} x_i \\ x_j \\ x_k \end{bmatrix}, \quad y = [0 \quad \xi \quad 1 - \xi] \begin{bmatrix} y_i \\ y_j \\ y_k \end{bmatrix}. \quad (6.4)$$

3. The transformation is **bijective**. This is valid only if the Jacobian matrix  $\mathbf{J}$  is not singular:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} x_j - x_i & y_j - y_i \\ x_k - x_i & y_k - y_i \end{bmatrix}. \quad (6.5)$$

This is equal to 0 only if the vertices are aligned.



Geometrically, we can see the  $\xi$ -variables as a local coordinate system. The process of generating meshes on surfaces is called **tesselation**. Reference elements are listed on Figure 6.7 and remark that for the 3D case, we have a system  $(\xi, \eta, \zeta)$ .

Figure 6.6

Order Dimension	Linear	Quadratic	Cubic
1D	$\xi = -1, 0, 1$	$\xi = -1, 0, 1, \frac{1}{2}$	$\xi = -1, -\frac{1}{3}, 0, \frac{1}{3}, 1$
2D (triangular)	$\xi = 0, 1, \eta = 0, 1$	$\xi = 0, 1, \eta = 0, 1, \sqrt{2}/2$	$\xi = 0, 1, \eta = 0, 1, \sqrt{2}/3, 2\sqrt{2}/3$
2D (quadrangular)	$\xi = -1, 1, \eta = -1, 1$	$\xi = -1, 1, \eta = -1, 1, \sqrt{2}/2, -\sqrt{2}/2$	$\xi = -1, 1, \eta = -1, 1, \sqrt{2}/3, -\sqrt{2}/3, 2\sqrt{2}/3, -2\sqrt{2}/3$
3D (tetrahedral)	$\xi = 0, 1, \eta = 0, 1, \zeta = 0, 1$	$\xi = 0, 1, \eta = 0, 1, \zeta = 0, 1$	$\xi = 0, 1, \eta = 0, 1, \zeta = 0, 1$
3D (hexahedral)	$\xi = -1, 1, \eta = -1, 1, \zeta = -1, 1$	$\xi = -1, 1, \eta = -1, 1, \zeta = -1, 1$	$\xi = -1, 1, \eta = -1, 1, \zeta = -1, 1$
3D (prismatic)	$\xi = 0, 1, \eta = 0, 1, \zeta = 0, 1$	$\xi = 0, 1, \eta = 0, 1, \zeta = 0, 1$	$\xi = 0, 1, \eta = 0, 1, \zeta = 0, 1$

Figure 6.7

## 6.4 Approximation on the reference element

Remind that we want to approximate a quantity of interest  $u(\mathbf{x})$ :

$$u(\mathbf{x}) = [N_1(\mathbf{x}) \quad N_2(\mathbf{x}) \dots] \begin{bmatrix} u_1 \\ u_2 \\ \vdots \end{bmatrix} = \mathbf{N}^e(\mathbf{x}) \mathbf{q}^e \quad (6.6)$$

This expression is in the real space, using the  $\mathcal{T}$  transformation:

$$u(\xi) = \bar{\mathbf{N}}^e(\xi) \mathbf{q}^e, \quad \mathcal{T}: \xi \rightarrow x(\xi) = \bar{\mathbf{N}}^e \mathbf{x}^e. \quad (6.7)$$

We have 4 properties:

1. the approx is **interpolant**:  $u(\xi_i) = u_i$ , related to  $N_j(\xi_i) = \delta_{ij}$ ;
2. **continuity in the element**: the  $N_j$  and their derivatives up to the order  $s$  must be continuous;
3. **continuity between elements**: the approx of  $u$  and its derivatives up to the order  $s$  must depend univocally on the nodal variables appearing on the boundary between the elements and nothing else;
4. when the size of the element tends to 0, the error on the approx and its derivatives vanish.

It is also possible to demonstrate that  $\sum_i N_i(\xi) = 1$ . Approx of  $C^0$ - or  $C^s$ -type if  $u$  or  $u$  and its derivatives up to the order  $s$  are continuous. Finally, if  $\bar{\mathbf{N}} = \mathbf{N}$  the elements are **isoparametric**. If not:

- **sub-parametric** if the order of polynomials of the geometric transformation is lower than for the shape functions;
- **super-parametric** if the order is higher.

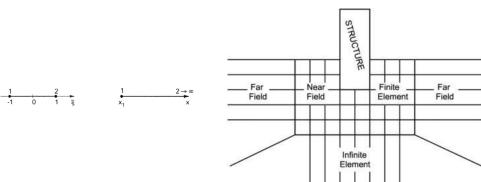


Figure 6.8

There can also be **infinite elements**. They can be devised in 1D by the following mapping:

$$x = x_1 + \alpha \frac{1 + \xi}{1 - \xi} \quad (6.8)$$

varying from  $x = x_1$  to  $\infty$  for  $\xi$  from -1 to 1. Used for infinite domains like the soil.

## 6.5 Construction of the shape function

This will be shown for the example of a four-node quadrangular element. We are searching for  $\mathbf{N}^e$  such that  $u^e(\mathbf{x}) = \mathbf{N}^e(\mathbf{x}) \mathbf{q}^e$ . First a polynomial basis must be set up with  $u(\xi) = \mathbf{p}(\xi)^T a$ , then by  $u(\xi) = \mathbf{p}(\xi)^T a = \mathbf{N}^e q^e$  we can retrieve the shape functions. A suitable polynomial is the one containing four terms and symmetric wrt  $\xi$  and  $\eta$ :

$$\mathbf{p}(\mathbf{x}) = [1 \quad \xi \quad \eta \quad \xi\eta]. \quad (6.9)$$

Then the interpolation:

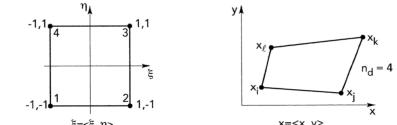


Figure 6.9

$$\mathbf{q}^e = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \begin{bmatrix} p_1(\xi_1) & p_2(\xi_1) & p_3(\xi_1) & p_4(\xi_1) \\ p_1(\xi_2) & p_2(\xi_2) & p_3(\xi_2) & p_4(\xi_2) \\ \vdots & & & \vdots \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \quad (6.10)$$

By inverting the matrix  $\mathbf{P}$ , we get the coefficients:  $\mathbf{a} = \mathbf{P}^{-1}\mathbf{q}^e$ . Finally, the shape functions can be obtained as follows:

$$u(\xi) = \mathbf{N}^e(\xi)\mathbf{q}^e = \mathbf{p}(\xi)^T \mathbf{a} = \mathbf{p}(\xi)^T \mathbf{P}^{-1} \mathbf{q}^e \quad \Rightarrow \mathbf{N}^e(\xi) = \mathbf{p}(\xi)^T \mathbf{P}^{-1}. \quad (6.11)$$

# Chapter 7

## Discretization by finite elements

The formulations we will use:

- **strong formulation:**

$$b_i \tau_{ij,j} = 0; \quad (7.1)$$

- **weak integral formulation:**  $\mathbf{u}$  solution of the elastic problem if and only if:

$$a(\mathbf{u}, \hat{\mathbf{u}}) - \varphi(\hat{\mathbf{u}}) = 0 \quad \forall \hat{\mathbf{u}} = 0 \text{ on } S_u \quad (7.2)$$

- **variational formulation:** // if and only if  $\delta\Pi(\mathbf{u})$  is minimum.

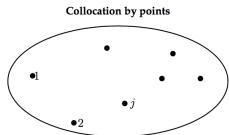
We will inject the discretized version of the displacement.

### 7.1 Weighted residual method

For the equilibrium equations the **residual**  $R_i(\mathbf{u})$  is given by:

$$R_i(\mathbf{u}) = b_i \tau_{ij,j}(\mathbf{u}). \quad (7.3)$$

#### 7.1.1 Collocations by points



To satisfy the equilibrium in the whole body  $V$ , one way is to make the residual vanish in a set of points distributed in the body  $R_i(\mathbf{u})^{(1)} = 0, \dots, R_i(\mathbf{u})^{(n_c)} = 0$ . This is not easily combined with finite elements and we have formally a system of  $n_c$  equations:

Figure 7.1

$$\int_V R_i \delta(x_j) dV = R_i|_{x_j} = 0. \quad (7.4)$$

#### 7.1.2 Collocation by sub-domains

Similarly to the previous point, but with sub-domains:

$$\int_V R_i \psi_j dV = \int_{V_j} R_i dV, \quad \psi_j = \begin{cases} 1 & \text{if } x \text{ belongs to } V^j \\ 0 & \text{if } x \text{ belongs to } V \neq V^j \end{cases} \quad (7.5)$$

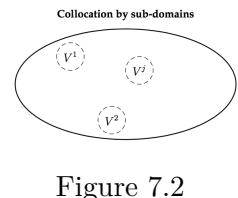


Figure 7.2

Here we get a system of  $n_d$  equations,  $n_d$  the number of sub-domains.

### 7.1.3 Least square method

This consist in minimizing the sum of the square of the residuals:

$$\min \int_V R_i^2 dV. \quad (7.6)$$

These last 3 methods cannot be easily implemented, unlike the next method (why do we see them then...)

### 7.1.4 Galerkin method

This requires the weak integral form and the discretization for the displacements. We will consider  $u$  as being the continuous displacement and  $u^h$  its finite elements approximation:

$$u^h(x) = N(x)q. \quad (7.7)$$

From seeking a continuous field  $u$  we have moved to search for a finite set of nodal values  $q$  (displacement of the nodes). Practically:

$$\text{in } 2D \rightarrow u^h(x) = \begin{bmatrix} u^h(x, y) \\ v^h(x, y) \end{bmatrix} \quad \text{in } 3D \rightarrow u^h(x) = \begin{bmatrix} u^h(x, y, z) \\ v^h(x, y, z) \\ w^h(x, y, z) \end{bmatrix} \quad (7.8)$$

Next, the linear stress tensor can be computed, since it is symmetric only 3 components in 2D and 6 in 3D are necessary:

$$\text{in } 2D \rightarrow \epsilon^h(x) = \begin{bmatrix} \epsilon_x^h(x, y) \\ \epsilon_y^h(x, y) \\ \gamma_{xy}^h(x, y) \end{bmatrix} = \begin{bmatrix} \partial_x & 0 \\ 0 & \partial_y \\ \partial_y & \partial_x \end{bmatrix} \begin{bmatrix} u_x^h(x, y) \\ v_y^h(x, y) \end{bmatrix} \quad (7.9)$$

and same way for the 3D case. In short this can be written as:

$$\epsilon^h(x) = Du^h(x) = \underbrace{DN(x)}_B q, \quad (7.10)$$

where D is the derivation matrix and B the matrix containing the derivatives of the shape functions. Since the material is homogeneous, isotropic and linear elastic, Hooke's law gives a linear relation between the strains and the stresses. If we store the stresses in a vector, they can be obtained via the **Hooke's matrix** in 3D:

$$H = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} = \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix} \quad (7.11)$$

Note that in 2D structures, due to the Poisson effect, we also have to take into account the z direction. 2 cases can occur:

- **plane stress state:** the loading and the stresses only occur on the  $(x, y)$ -plane, and the dispacements are free in the z direction:  $\sigma_z = 0, \epsilon_z \neq 0$ .
- **plane strain state:** the displacements in z directions are blocked, giving birth to stresses:  $\sigma_z \neq 0, \epsilon_z = 0$ .

We have to consider a different Hooke's matrix for each cases:

$$H_{\text{plane stress}} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \quad H_{\text{plane strain}} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}. \quad (7.12)$$

These can be shorten in:

$$\tau^h(x) = H\epsilon^h(x) = HB(x)q. \quad (7.13)$$

The strain energy  $W_V$  can be written as the dot product of the stress and strain vectors:

$$W_V = \frac{1}{2}\tau_{ij}\epsilon_{ij} \Rightarrow W_V^h = \frac{1}{2}\epsilon^{hT}\tau^h. \quad (7.14)$$

Let's now substitute these lasts into the weak formulation:

$$\Rightarrow \int_V \tau_{ij}\hat{\epsilon}_{ij} dV - \int_V b_i\hat{u}_i dV - \int_{S_t} \bar{t}_i^{(n)}\hat{u}_i dS = 0 \quad \forall \hat{u} = 0 \text{ on } S_u. \quad (7.15)$$

The virtual displacements  $\hat{u}_i$  have not a direct physical meaning, they are trial functions to verify governing equations. They can be approximated by the same shape function as the real displacements:

$$\hat{u}^h(x) = N(x)\hat{q}. \quad (7.16)$$

Instead of exploring all the continuous trial functions, we have the discrete points  $\hat{q}$ :

$$\begin{aligned} &\Rightarrow \int_V \epsilon^{hT}\hat{\tau}^h dV - \int_V \hat{q}^T N^T b dV - \int_{S_t} \hat{q}^T N^T \bar{t}^{(n)} dS = 0 \quad \forall \hat{q} = 0 \text{ on } S_u \\ &\Leftrightarrow \hat{q}^T \left( \underbrace{\int_V B^T H B dV}_K q - \underbrace{\int_V N^T b dV}_{f^V} - \underbrace{\int_{S_t} N^T \bar{t}^{(n)} dS}_{f^S} \right) = 0 \quad \forall \hat{q} = 0 \text{ on } S_u \\ &\Leftrightarrow Kq = f^V + f^S = f. \end{aligned} \quad (7.17)$$

We see that if the equation holds for any  $\hat{q}$  we get the canonical form:

$$\boxed{\mathbf{K}\mathbf{q} = \mathbf{f}} \quad (7.18)$$

where  $K$  is a  $Q$  by  $Q$  matrix and  $q, f$  are  $Q$ -sized vectors, with  $Q$  being the number of degrees of freedom, equal to the number of nodes multiplied by the number of degrees of freedom per node.  $K$  is the stiffness matrix,  $f$  are the nodal forces and  $q$  the nodal displacements. The only unknown is  $q$ .

## 7.2 Ritz analysis methods

This one bases on the stationarity of the  $\Pi$  functional. The discretized version of  $\Pi(u) = a(u, u) - \varphi(u)$  is:

$$\begin{aligned}\Pi(u^h) &= \frac{1}{2}q^T \left( \int_V B^T H B dV \right) q - q^T \int_V N^T b dV - q^T \int_{S^t} N^T \bar{t}^{(n)} dS \\ &= \frac{1}{2}q^T K q - q^T f^V - q^T f^S \Rightarrow \delta\Pi(u) = \frac{1}{2}\delta q^T K q + \frac{1}{2}q^T K \delta q - \delta q^T f^V - \delta q^T f^S.\end{aligned}\tag{7.19}$$

This derivative vanishes for all  $\delta q$  provided  $Kq = f^V + f^S = f$ , as the previous case.

### 7.3 Properties of the finite element solution

We know that the finite element  $u^h$  and the theoretical continuous solution satisfy:

$$a(u^h, \hat{u}^h) - \varphi(\hat{u}^h) = 0 \quad a(u, \hat{u}^h) - \varphi(\hat{u}^h) = 0 \quad \forall \hat{u} = 0 \text{ on } S_u.\tag{7.20}$$

By substracting the fist from the second we get:

$$a(u - u^h, \hat{u}^h) = a(e^h, \hat{u}^h) = 0 \quad \forall \hat{u} = 0 \text{ on } S_u,\tag{7.21}$$

where  $e$  is the error. This last equation expresses that the finite element solution is the best we can find in the finite element function space. Furthermore:

$$a(u, u) = a(u^h + e^h, u^h + e^h) = a(u^h, u^h) + \underbrace{2(u^h, e^h)}_{=0} + a(e^h, e^h),\tag{7.22}$$

where the  $a$  functional is equal to twice the strain energy  $W$ , stating that the strain energy of the theoretical solution is an upper bound for the finite element solution.

# Chapter 8

## 2D elements in plane stress and plane strain

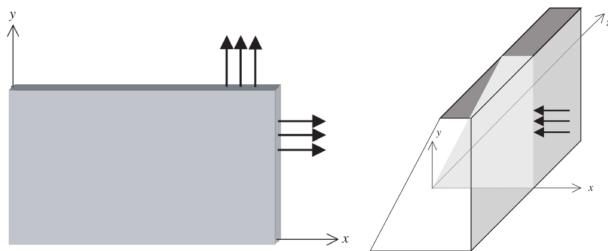


Figure 8.1

### 8.1 Triangular elements

#### 8.1.1 TRIM-3

The first element is a triangle with three nodes on the vertex of the triangle. We have 3 degree of freedom for each nodes meaning that 6 degree of freedom characterize this element:

$$q^e = [u_1 \ v_1 \ u_2 \ v_2 \ u_3 \ v_3]^T. \quad (8.1)$$

TRIM-3 means triangle, membrane and 3 nodes. Notice that only linear displacements can be performed since we have 2 nodes per side. We will also assume that:

- the nodal displacement following x-axis ( $u^e(x)$ ) only depends on the x nodal displacement:  $u_i$ ;
- same for y-axis
- same shape functions are used for u and v.

From these hypothesis, we can approximate the displacement as:

$$u^e(x) = N^e(x)q^e = \begin{bmatrix} N_1(x) & 0 & N_2(x) & 0 & N_3(x) & 0 \\ 0 & N_1(x) & 0 & N_2(x) & 0 & N_3(x) \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix} \quad (8.2)$$

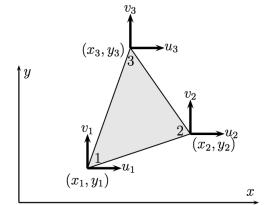


Figure 8.2

Enforcing the approximation to be interpolant at the nodes we have 6 additional constraints:

$$u^e(x_j) = u_i N_i(x_j) = u_i \quad v^e(x_j) = v_i N_i(x_j) = v_i \quad \Rightarrow N_i(x_j) = \delta_{ij}. \quad (8.3)$$

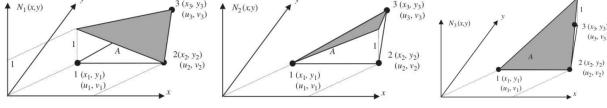


Figure 8.3

( $x_j, y_j, 0$ ) for  $j \neq i$ . For  $N_1(x)$  for example we have:

$$\begin{vmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x \\ y_1 & y_2 & y_3 & y \\ 1 & 0 & 0 & N_1(x) \end{vmatrix} = 0 \quad \Rightarrow N_1(x) = -\frac{1}{2\Delta}[(y_3 - y_2)(x - x_2) - (x_3 - x_2)(y - y_2)] \quad (8.4)$$

With  $\Delta = \frac{1}{2} \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix}$  which is the signed area of the element. Note that the numbering of the nodes has an impact. The numbering should be the same for all the elements. From the displacement, the strains can be obtained:

$$\epsilon^e(x) = Du^e(x) = DN^e(x)q^e = B(x)q^e. \quad (8.5)$$

The TRIM-3 element is the rare case where the  $B(x)$  ends up as a constant, not depending on the actual position  $x$ :

$$B(x) = -\frac{1}{2A^e} \begin{bmatrix} y_3 - y_2 & 0 & y_1 - y_3 & 0 & y_2 - y_1 & 0 \\ 0 & -(x_3 - x_2) & 0 & -(x_1 - x_3) & 0 & -(x_2 - x_1) \\ -(x_3 - x_2) & y_3 - y_2 & -(x_1 - x_3) & y_1 - y_3 & -(x_2 - x_1) & y_2 - y_1 \end{bmatrix} \quad (8.6)$$

Finally, as  $B$  is a constant, the integral for the stiffness matrix  $K$  gives:

$$K = B^T H B d^e A^e \quad (8.7)$$

where  $d^e$  is the thickness and  $A^e$  the area of the volume  $V^e$  of the element. The forces have to be put in nodal values, from the definition:

$$f_{i,x}^{e,V} = \int_{V^e} b_x dV = \frac{1}{3} b_x d^e A^e \quad f_{i,y}^{e,V} = \int_{V^e} b_y dV = \frac{1}{3} b_y d^e A^e \quad (8.8)$$

For the surface forces, we have to consider the forces integrated on each side of the element:

$$f_{i,x}^{e,S} = \int_{S_t} N_i t_x^{(n)} dS = \frac{1}{2} d^e l_{side\ i-j} t_x^{(n)} = f_{j,x}^{e,S}. \quad (8.9)$$

As mentioned, the TRIM-3 element is a particular case where the  $B$  matrix is constant. While the displacement are continuous across the elements, the strains and stresses are constant on each elements and so discontinuous across the elements.  $K^e$  is obtained by matrix multiplication, without integration.

The nodes functions are represented here, we see that it has to be 1 on the  $i$ -th node and 0 at the others, linearly. The analytical expression can be obtained by considering it co-planar to  $(x_i, y_i, 1)$  and  $(x_j, y_j, 0)$  for  $j \neq i$ . For  $N_1(x)$  for example we have:

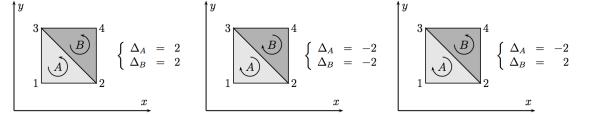


Figure 8.4

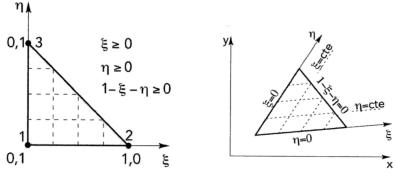


Figure 8.5

$$x = \sum_{i=1}^3 N_i(\xi, \eta) x_i \quad y = \sum_{i=1}^3 N_i(\xi, \eta) y_i \quad (8.10)$$

The goal is to replace the stiffness calculation in real space by its calculation on the reference element:

$$\int_{V^e} B(x, y)^T H B(x, y) dx dy dz \Rightarrow \int_{\xi} \int_{\eta} \int_z B(\xi, \eta)^T H B(\xi, \eta) \det(J) d\xi d\eta dz \quad (8.11)$$

for a constant thickness in the z-axis, we can replace  $\int_z dz$  by  $d^e$  directly. The derivatives of the shapes functions wrt x, y involves the derivatives wrt  $\xi, \eta$  by chain:

$$\begin{aligned} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix} &= \underbrace{\begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix}}_J \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n_e} \frac{\partial N_i(\xi, \eta)}{\partial \xi} x_i & \sum_{i=1}^{n_e} \frac{\partial N_i(\xi, \eta)}{\partial \xi} y_i \\ \sum_{i=1}^{n_e} \frac{\partial N_i(\xi, \eta)}{\partial \eta} x_i & \sum_{i=1}^{n_e} \frac{\partial N_i(\xi, \eta)}{\partial \eta} y_i \end{bmatrix} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} \\ &\Rightarrow \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = J^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix}. \end{aligned} \quad (8.12)$$

Practically, the computation of  $K^e$  is not performed analytically (ouf) but numerically, see later. There are 2 ways to improve the accuracy, increasing the number of nodes or increasing the order of the approximation. To implement the **quadratic** approximation required from the second method, we need the six-node element.

### 8.1.2 TRIM-6

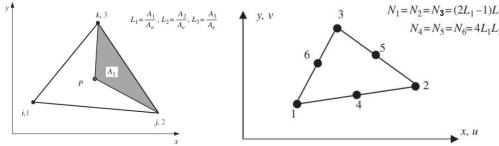


Figure 8.6

at node  $i$  and  $L_i = 0$  for nodes  $i \neq j$  (figure: left). We see on figure: right that the shape functions in TRIM-6 have an easy expression too.

From a similar integration to what we have seen, the body and contact forces are given by:

$$\begin{cases} f_{i,x}^{e,V} = 0 & i = 1, 2, 3 \\ f_{i,x}^{e,V} = \frac{1}{3} b_x d^e A^e & i = 4, 5, 6 \end{cases} \quad \begin{cases} f_{i,x}^{e,S} = \frac{1}{6} t_x^{(n)} l_{side i-j} d^e = f_{j,x}^{e,S} \\ f_{k,x}^{e,S} = \frac{4}{6} t_x^{(n)} l_{side i-j} d^e \quad k \text{ middle of side } i-j \end{cases} \quad (8.13)$$

We see that for the quadratic form, the forces are not equally distributed anymore, the weight is dictated by the position.

## 8.2 Quadrangular elements

### 8.2.1 REM-4

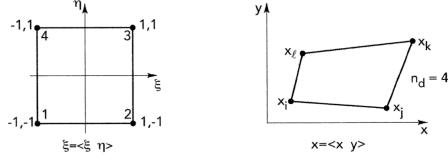


Figure 8.7

$$\begin{aligned} N_1(\xi, \eta) &= \frac{1}{4}(1 - \xi)(1 - \eta), & N_2(\xi, \eta) &= \frac{1}{4}(1 + \xi)(1 - \eta), \\ N_3(\xi, \eta) &= \frac{1}{4}(1 + \xi)(1 + \eta), & N_4(\xi, \eta) &= \frac{1}{4}(1 - \xi)(1 + \eta) \end{aligned} \quad (8.14)$$

Note that the shape functions are linear on each side but note within the element since we have now a term in  $\xi\eta$ . The body and contact forces are:

$$f_{i,x}^{e,V} = \frac{1}{4}b_x d^e A^e \quad f_{i,x}^{e,S} = f_{j,x}^{e,S} = \frac{1}{2}t_x^{(n)} l_{side\ i-j} d^e \quad (8.15)$$

for constant  $d^e$  and  $t_x^{(n)}$ .

### 8.2.2 REM-8 and REM-9

Better accuracy can be reached by increasing the number of nodes. However, since only 8 nodes are implied, the corresponding polynomial approximation does not contain all the bi-quadratic polynomial terms omitting  $x^2y^2$ . Elements containing incomplete polynomial basis are called **Serendipity elements**.

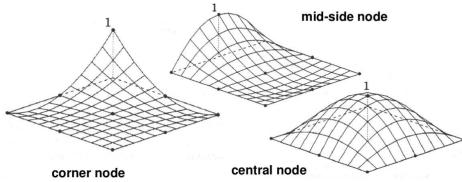


Figure 8.9

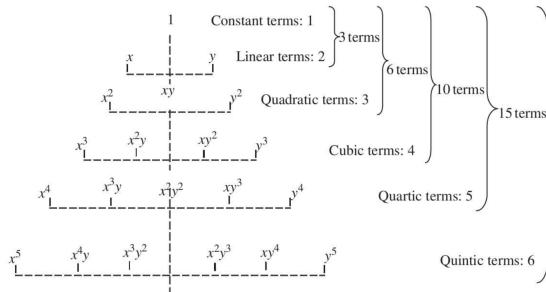


Figure 8.10

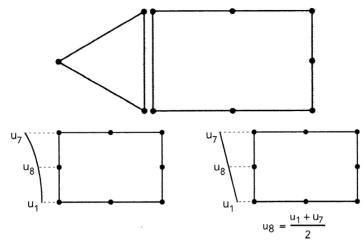
To get a complete bi-quadratic approximation, we add a node at the center (REM-9). The shape functions still satisfy the condition  $N_j(x_i) = \delta_{ij}$ . Such element is called **Lagrange element**. The full basis of polynomials (pascal triangle) and the forces are depicted on Figure 8.10 and Figure 8.11.

$$\begin{aligned} \rightarrow \text{REM-8 : } & \left\{ \begin{array}{ll} f_{i,x}^{e,V} &= -1/12 b_x A^e d^e & \text{for vertices} \\ f_{i,x}^{e,V} &= 1/3 b_x A^e d^e & \text{for middle nodes} \\ f_{i,x}^{e,S} &= 1/6 t_x^{(n)} l_{side\ i-j} d^e & \text{for vertices} \\ f_{k,x}^{e,S} &= 4/6 t_x^{(n)} l_{side\ i-j} d^e & \text{for } k \text{ the middle node on } l_{side\ i-j} \\ f_{i,x}^{e,V} &= -1/36 b_x A^e d^e & \text{for vertices} \\ f_{i,x}^{e,V} &= 4/36 b_x A^e d^e & \text{for middle nodes} \\ f_{i,x}^{e,S} &= 16/36 b_x A^e d^e & \text{for the central node} \\ f_{i,x}^{e,S} &= 1/6 t_x^{(n)} l_{side\ i-j} d^e & \text{for vertices} \\ f_{k,x}^{e,S} &= 4/6 t_x^{(n)} l_{side\ i-j} d^e & \text{for } k \text{ the middle node on } l_{side\ i-j} \end{array} \right. \\ \rightarrow \text{REM-9 : } & \end{aligned}$$

Figure 8.11

Remark that the central node does not intervene when assembled to other elements and can be extracted from the stiffness relations (**condensation** of central node).

### 8.3 Connecting elements of different types



Different elements can be connected if the continuity between the elements is ensured. For example, TRIM-3 and REM-4 is ok because they have both continuity on their side. Same for TRIM-6, REM-8 and REM-9. Linear and quadratic can be assembled but by adding a linearity constraint.

Figure 8.12

# Chapter 9

## Assembly and boundary conditions

### 9.1 Assembly

Each element is modeled by means of a local or elementary stiffness relation  $K^e q^e = f^e$ . The assembly of the elements constituting the whole domain V has two purposes:

- **compatibility constraint:** coherence of the displacements, a node appertaining to two element must undergo the same displacement;
- **equilibrium constraints:** equilibrium of the structure, eliminating the contact forces between the elements.

Element	Connecting nodes
1	{1, 3, 4}
2	{1, 4, 2}
3	{2, 5}
4	{3, 6, 7, 4}
5	{4, 7, 8, 5}

Figure 9.1

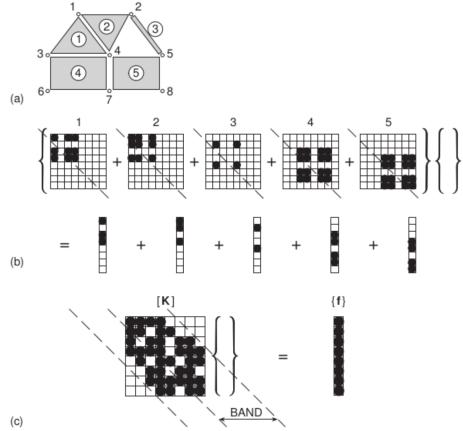


Figure 9.2

The general pattern of the assembly procedure is represented on Figure 9.1 and Figure 9.2. The stiffness relation can be decomposed:

$$K^e q^e = f^e = f^{V,e} + f^{S_{ext},e} + f^{S_{ext},e} + f^{S_{neighb},e} \quad (9.1)$$

where we see the external forces and the one due to the neighboring elements. Indeed, the neighboring forces vanish due to action reaction principle when the whole structure is considered.

### 9.2 Boundary conditions

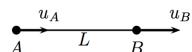


Figure 9.3

We have our matrix  $K$  but solving  $Kq = f$  as it is would not work, because  $K$  is not invertible. To understand this behavior let's take the example of a bar in tension. The system is defined by the displacements on the extreme

nodes. However, we observe that for the same strain level ( $\epsilon_x = (u_B - u_A)/L$ ), we can have an infinite number of couple  $(u_A, u_B)$ . The system is **underdetermined**. Physically, the rigid body motion has not been blocked yet. We have to add the supports (boundary conditions).

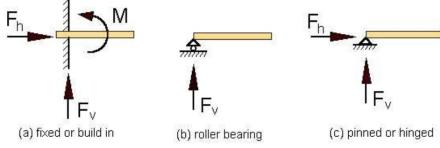


Figure 9.4  
supports represented on Figure 9.4.

In 2D we have 3 rigid body modes, 2 translations 1 rotation. In 3D we have 6, 3 translations and 3 rotations. The underdeterminacy provokes the singularity of the stiffness matrix. This doesn't depend on the external forces, we only have that the solution found by solving the stiffness relation is not unique. There are 3 main types of

### 9.2.1 Direct method

Assuming a given degree of freedom  $i$   $q_i = \bar{q}_i$ , if one direction is blocked and we apply a displacement, an unknown reaction force  $R_i$  will appear in addition to external nodal forces:

$$\begin{bmatrix} K_{11} & \dots & K_{1i} & \dots & K_{1Q} \\ \vdots & & \vdots & & \vdots \\ K_{i1} & \dots & K_{ii} & \dots & K_{iQ} \\ \vdots & & \vdots & & \vdots \\ K_{Q1} & \dots & K_{Qi} & \dots & K_{QQ} \end{bmatrix} \begin{bmatrix} q_1 \\ \vdots \\ \bar{q}_i \\ \vdots \\ q_Q \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_i + R_i \\ \vdots \\ f_Q \end{bmatrix}. \quad (9.2)$$

The displacement in the  $i$ -th degree is known but not the force. For every other line  $j$  we have:

$$K_{j1}q_1 + \dots + K_{ji}\bar{q}_i + \dots + K_{jQ}q_Q = f_j \Rightarrow K_{j1}q_1 + \dots + K_{jQ}q_Q = f_j - K_{ji}\bar{q}_i. \quad (9.3)$$

By replacing the known displacements and discarding the corresponding equations we get:

$$\begin{bmatrix} K_{11} & \dots & / & \dots & K_{1Q} \\ \vdots & & / & & \vdots \\ / & / & / & / & / \\ \vdots & & / & & \vdots \\ K_{Q1} & \dots & / & \dots & K_{QQ} \end{bmatrix} \begin{bmatrix} q_1 \\ \vdots \\ / \\ \vdots \\ q_Q \end{bmatrix} = \begin{bmatrix} f_1 - K_{1i}\bar{q}_i \\ \vdots \\ / \\ \vdots \\ f_Q - K_{Qi}\bar{q}_i \end{bmatrix}. \quad (9.4)$$

We see thus that the system is reduced to a  $(Q - m)$ sized matrix where  $Q$  is the number of degrees of freedom. Once the unknown displacements are known, we can find the reaction:

$$R_i = \left( \sum_{j=1}^Q K_{ij}q_j \right) - f_i. \quad (9.5)$$

### 9.2.2 Penalty method

Another approach consists in penalizing the  $i$ -th imposed displacement by a very large number  $Z$  ( $\approx 10^{20}$ ).  $K_{ii}$  is replaced by  $K_{ii} + Z$  and  $f_i$  is replaced by  $f_i + Z\bar{q}_i$ . The system becomes:

$$\begin{bmatrix} K_{11} & \dots & K_{1i} & \dots & K_{1Q} \\ \vdots & & \vdots & & \vdots \\ K_{i1} & \dots & K_{ii} + Z & \dots & K_{iQ} \\ \vdots & & \vdots & & \vdots \\ K_{Q1} & \dots & K_{Qi} & \dots & K_{QQ} \end{bmatrix} \begin{bmatrix} q_1 \\ \vdots \\ \bar{q}_i \\ \vdots \\ q_Q \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_i + Z\bar{q}_i \\ \vdots \\ f_Q \end{bmatrix}. \quad (9.6)$$

where the i-th line is:  $Zq_i + (\sum_{j=1}^Q K_{ij}q_j) = Z\bar{q}_i$ , which is almost equivalent to  $q_i = \bar{q}_i$  for  $Z \gg$ . The reactions are given by:

$$R_i = Z(\bar{q}_i - q_i). \quad (9.7)$$

The only limitation of this method is the round-off errors when  $\bar{q}_i \neq 0$ . For example for a 16 digits precision  $q_i = \bar{q}_i - 10^{-20} \approx \bar{q}_i$  so  $R_i = 0$  while for  $\bar{q}_i = 0$  we have good results.

# Chapter 10

## Numerical integration

### 10.1 Exploiting isoparametric properties

Except for the TRIM-3 element we already discussed, the other elements require an integration over the element with varying shape. Here is the efficiency of the isoparametric elements:

- instead of integrating on the real element, it will be done on a parent element;
- due to the presence of the inverse of the Jacobian, the integration will be done numerically:

$$K^e = \int_{V^e} B(x)^T H B(x) dV \quad \Rightarrow K^e = \int_{\xi} \int_{\eta} \int_{\zeta} B(x(\xi))^T H B(x(\xi)) \det(J(\xi, \eta)) d\xi d\eta d\zeta \quad (10.1)$$

where the  $B(x(\xi))$  contain the derivative of the shape functions wrt  $x, y, z$ .

For the integration process, let's consider a constant thickness  $d^e$ . The developments of chapter 8 will be used. Remind that the derivatives of  $N(x, y)$  respect the chain rule:

$$\begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix}. \quad (10.2)$$

Knowing that the coordinates are related as:

$$x(\xi, \eta) = \sum_{i=1}^{n_e} N_i(\xi, \eta) x_i, \quad y(\xi, \eta) = \sum_{i=1}^{n_e} N_i(\xi, \eta) y_i \quad (10.3)$$

the derivatives are given by:

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = \underbrace{\begin{bmatrix} \sum_{i=1}^{n_e} \frac{\partial N_i}{\partial \xi} x_i & \sum_{i=1}^{n_e} \frac{\partial N_i}{\partial \xi} y_i \\ \sum_{i=1}^{n_e} \frac{\partial N_i}{\partial \eta} x_i & \sum_{i=1}^{n_e} \frac{\partial N_i}{\partial \eta} y_i \end{bmatrix}}_{J^{-1}}^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix}. \quad (10.4)$$

The derivatives wrt  $\xi, \eta$  are easy to compute (see p.83 of the syllabus for REM-4). But the Jacobian containing terms in  $\xi, \eta, x_j, y_j$ , the analytical integration is no longer possible.

### 10.2 Numerical integration

Numerical integration methods consist in replacing the calculation of an integral by a weighted sum of function values at specified points. In 1D the integral of a function  $f(x)$  is estimated as:

$$\int_{-1}^1 f(x) dx \approx \sum_{k=1}^K w_k f(x_k), \quad (10.5)$$

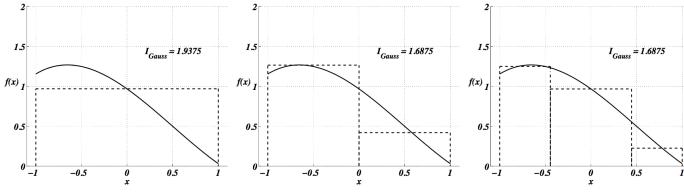


Figure 10.1  
applied on the REM-4 gives:

$$\begin{aligned} K^e &= \int_{-1}^1 \int_{-1}^1 B(\xi, \eta)^T H B(\xi, \eta) d^e \det(J(\xi, \eta)) d\xi d\eta \\ &\approx \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} B(\xi_{k_1}, \eta_{k_2})^T H B(\xi_{k_1}, \eta_{k_2}) d^e \det(J(\xi_{k_1}, \eta_{k_2})) w_{k_1} w_{k_2}. \end{aligned} \quad (10.6)$$

Note that the external forces can also be computed like that:

$$f^{V,e} = \int_{V^e} N^T b dV \approx \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} N(\xi_{k_1}, \eta_{k_2})^T B d^e \det(J(\xi_{k_1}, \eta_{k_2})) w_{k_1} w_{k_2}. \quad (10.7)$$

it's a bit different for surfaces forces which are applied on faces defined as  $\xi, \eta = cst$  or  $\xi = cst, \eta$ . In the first case we have:

$$\begin{aligned} ds &= \sqrt{\left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2} d\xi, \\ \Rightarrow f^{S,e} &= \int_{S_d^e} N^T \bar{t}^{(n)} dS \approx \sum_{k_1=1}^{K_1} N(\xi_{k_1})^T \bar{t}^{(n)} d^e \sqrt{\left(\frac{\partial x}{\partial \xi}\right)_{\xi_{k_1}}^2 + \left(\frac{\partial y}{\partial \xi}\right)_{\xi_{k_1}}^2} \end{aligned} \quad (10.8)$$

and similarly for  $\eta$ .

## 10.3 Selection of the quadrature order

From the continuous form to the finite element approximation we have already 2 errors:

- error on the geometry, if the mesh does not exactly coincide with the real geometry;
- discretization error, the solution is not searched from all the functions but only some discrete subset.

Another one is the integration error. This is related to the number of Gauss points.

### 10.3.1 Matrix singularity due to numerical integration

	Linear		Quadratic	
	Degree of freedom	Independent relation	Degree of freedom	Independent relation
(a)	$4 \times 2 - 3 = 5 > 1 \times 3 = 3$ singular		$2 \times 8 - 3 = 13 > 4 \times 3 = 12$ singular	
(b)	$6 \times 2 - 3 = 9 > 2 \times 3 = 6$ singular		$13 \times 2 - 3 = 23 < 8 \times 3 = 24$	
(c)	$25 \times 2 - 18 = 32 < 16 \times 3 = 48$		$48 \times 2 = 96 < 64 \times 3 = 192$	

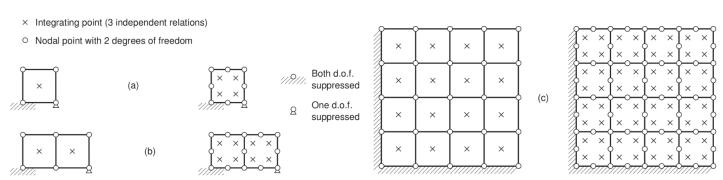


Figure 10.2

Figure 10.3

The insufficient number of Gauss points is related to the stiffness matrix conditioning. A few examples with 2 degrees of freedom per nodes are depicted on Figure 10.3. At each integration point 3 strain relations ( $\epsilon_x, \epsilon_y, \gamma_{xy}$ ) gives a total variable number of  $3N_{integ}$ . The number of unknowns is  $2N - r$  where  $N$  is the number of nodes and  $r$  the number of restrained dof. Figure 10.2 collects the balance between equations and unknowns for all the examples and shows the importance of sufficient  $N_{integ}$ .

### 10.3.2 Detection of skewed elements

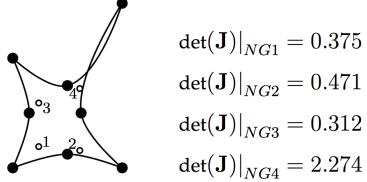


Figure 10.4

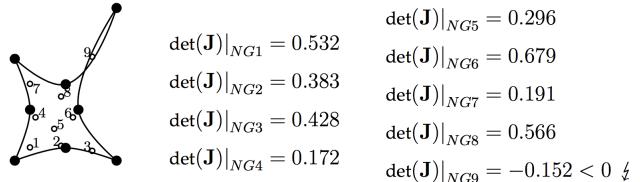


Figure 10.5

Since node coordinates are independent parameters, some configurations can lead to distorted or **skewed** elements. Since the Jacobian is computed by adding the contribution on all Gauss point, the configuration on Figure 10.4 does not allow to detect skewness, while on Figure 10.5 this is possible as the last determinant of the Jacobian matrix is negative.

### 10.3.3 Full vs. reduced integration

How many Gauss point do we need? As seen  $K$  points allow to calculate exactly a  $(2K-1)$ -order polynomial. So we should select the number of integration point in function of the polynomial order, but the inverse of Jacobian matrix hinder an exact integration. A reasonable choice is a quadrature order that integrates exactly for a rectangular or straight side triangular element.

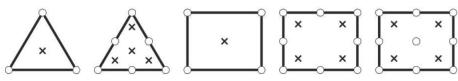


Figure 10.6

Element	Degree of $N_i$ in $\xi$ and $\eta$	Degree of $\mathbf{B}$ in $\xi$ and $\eta$	Degree of the integrants of $\mathbf{K}^e$ in $\xi$ and $\eta$	Full integration
REM-4	$1 \times 1$	$1 \times 1$	$2 \times 2$	$2 \times 2$ points
REM-8, REM-9	$2 \times 2$	$2 \times 2$	$4 \times 4$	$3 \times 3$ points

Figure 10.7

On the figures above we can see the number of Gauss points necessary to do the **exact integration**.

Element	Degree of $N_i$ in $\xi$ and $\eta$	Degree of $\mathbf{B}$ in $\xi$ and $\eta$	Degree of the integrants of $\mathbf{K}^e$ in $\xi$ and $\eta$	Reduced integration
REM-4	$1 \times 1$	$0 \times 1$ and $1 \times 0$	$1 \times 1$	$1 \times 1$ points
REM-8, REM-9	$2 \times 2$	$1 \times 2$ and $2 \times 1$	$3 \times 3$	$2 \times 2$ points

Figure 10.8

The **minimum quadrature order** is based on the guarantee of reproducing a constant strain when the element size tends to zero. In this case the order of integration should only be sufficient to assess the element volume like:

$$\int_{V^e} \det(J) d^e \mathbf{d}\xi \mathbf{d}\eta. \quad (10.9)$$

This process is called **reduced integration** and is shown on Figure 10.8. But this can lead to some problems like the stiffness singularity. Indeed, the strain field can be such to vanish at some integration point and cause singularity. Sometimes, this mechanism is compatible with adjacent elements and can lead to the singularity of the global matrix.

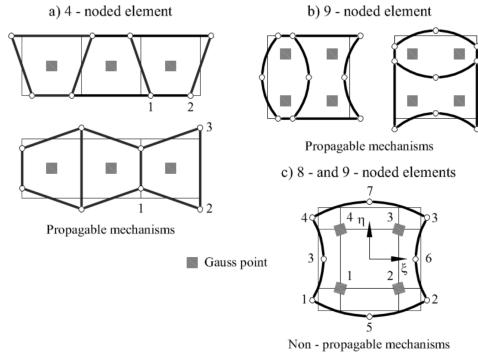


Figure 10.9

Since the rank of the product of matrices is the rank of the smallest one and since we have one term in the quadrature ( $NG = 1$ ), the rank of  $K$  is 3, while it should be equal to 5 in 2D (8 - 3 (rigid body motion) = 5). Reduced integration must be used with care.

But reduced integration is useful when minimum quadrature points coincide with the accurate points for the computation of strains and stresses. It can be shown that:

- the displacements are best sampled on the nodes of the element;
- for strains and stresses the best accuracy is obtained at the Gauss points, called **Barlow points** and the phenomenon is **superconvergence**.

As example consider the reduced one point in the REM-4 and the 2 mechanisms induced. The mechanisms in (c) are called Hourglass modes. They correspond to non-zero displacements endowed with zero energy. Mathematically with one Gauss point we have for a REM-4:

$$\underbrace{K^e}_{8 \times 8} = \underbrace{B(\xi_1, \eta_1)^T}_{8 \times 3} \underbrace{H}_{3 \times 3} \underbrace{B(\xi_1, \eta_1)}_{3 \times 8} \det(J(\xi_1, \eta_1)) d^e w_1 w_1. \quad (10.10)$$

# Chapter 11

## System resolution

### 11.1 Properties of the stiffness matrix

We have the assembly, the BC are applied, we are ready to solve the system:  $Kq = f$ .  $K$  is a symmetric matrix since:

$$K = \int_V B^T H B dV \quad (11.1)$$

where  $H$  is symmetric.  $K$  is positive definite after application of BC, the strain energy if  $q$  is non-zero is:

$$W = \frac{1}{2} q^T K q > 0. \quad (11.2)$$

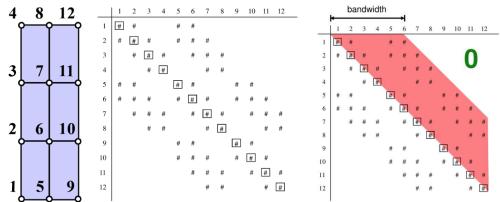


Figure 11.1

decrease the computation time!

Finally  $K$  is **sparse**, this means that the non-zero terms appear where the degrees of freedom are connected. The matrix is concentrated around the **bandwidth**, the narrowest zone parallel to the main diagonal for which all the terms vanish outside this zone. In addition,  $K$  is assumed to be of order  $n$ , full rank, invertible and non-singular. It is important for the matrix to be sparse, so that we significantly

### 11.2 Algorithms for solving linear systems

The main approaches are summarized on the figure. Direct methods are based on the bandwidth or the **frontwidth** which consists in progressing through the domain by Gauss elimination.

#### 11.2.1 Direct methods

It is composed of a single large step, works for any invertible matrix and provides the exact solution. Costly operations:  $O_{2D}(n^2), O_{3D}(n^{7/3})$ , and high memory usage. But it is faster for small matrices. It destroy the sparsity of the initial matrix. The Gauss elimination is used, we extract one unknown in function of the others and replace

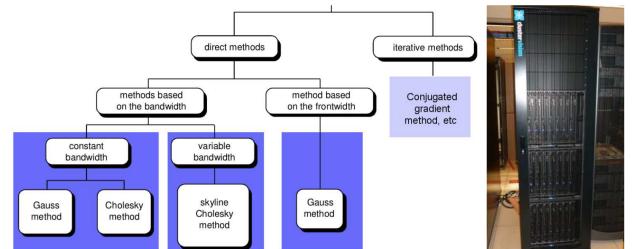


Figure 11.2

it in other equations to get a n-1 matrix, and we repeat this process till finding one unknown. The process is then repeated backward to compute all the unknowns. LU decomposition is represented in the next slides:

LU decomposition 16

$$[K] = [L][U]$$

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix}}_{[L]} \underbrace{\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}}_{[U]}$$

LU is called triangular decomposition, let us assume that this was done for now.

Step 1 – forward elimination

$$[L]\{y\} = \{f\}$$

LU decomposition 21

Step 1 – forward elimination

$$[L]\{y\} = \{f\}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \left\{ \begin{array}{l} y_1 \\ y_2 \\ y_3 \end{array} \right\} = \left\{ \begin{array}{l} f_1 \\ f_2 \\ f_3 \end{array} \right\}$$

Recursive formula:

$$y_i = f_i - \sum_{i=1}^{i-1} l_{ij} y_j$$

$$\begin{cases} y_1 = f_1 \\ y_2 = f_2 - l_{21} y_1 \\ y_3 = f_3 - l_{31} y_1 - l_{32} y_2 \end{cases}$$

Figure 11.3

Figure 11.4

LU decomposition 26

Step 2 – backward substitution

$$[U]\{q\} = \{y\}$$

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \left\{ \begin{array}{l} q_1 \\ q_2 \\ q_3 \end{array} \right\} = \left\{ \begin{array}{l} y_1 \\ y_2 \\ y_3 \end{array} \right\}$$

Recursive formula:

$$q_i = \frac{y_i - \sum_{j=i+1}^n u_{ij} a_j}{u_{ii}}$$

$$\begin{cases} q_3 = \frac{y_3}{u_{33}} \\ q_2 = \frac{y_2 - u_{23} q_3}{u_{22}} \\ q_1 = \frac{y_1 - u_{12} q_2 - u_{13} q_3}{u_{11}} \end{cases}$$

LU decomposition 26

Question: how to do the triangular decomposition?  $[K] = [L][U]$

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}$$

$$\begin{cases} k_{11} = u_{11} \\ k_{12} = u_{12} \\ k_{21} = l_{21} u_{11} \\ k_{22} = l_{21} u_{12} + l_{22} u_{22} \\ k_{31} = l_{31} u_{11} \\ k_{32} = l_{31} u_{12} + l_{32} u_{22} \\ k_{33} = l_{31} u_{13} + l_{32} u_{23} + u_{33} \end{cases}$$

Figure 11.5

Figure 11.6

LU decomposition 37

Question: how to do the triangular decomposition?

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

$$k_{ji} = \sum_{m=1}^{i-1} l_{jm} u_{mi}$$

$$l_{jj} = 1 \quad l_{ji} = \frac{k_{ji} - \sum_{m=1}^{i-1} l_{jm} u_{mi}}{u_{ii}}$$

$$u_{jj} = k_{jj} - \sum_{m=1}^{j-1} l_{jm} u_{mj}$$

$$u_{ij} = k_{ij} - \sum_{m=1}^{i-1} l_{im} u_{mi} \text{ for } i = 1..j-1$$

LU decomposition 38

**Remarks**  
Triangular decomposition is costly for a large number of dof

$$k_{ji} = \sum_{m=1}^{i-1} l_{jm} u_{mi}$$

If  $u_{ii} = 0$  rearrange  $[K]$     $l_{ji} = \frac{k_{ji} - \sum_{m=1}^{i-1} l_{jm} u_{mi}}{u_{ii}}$  pivot

If pivot is negative permutation of lines or columns is required.

Computational cost  $\approx 2n^2$

Figure 11.7

Figure 11.8

### 11.2.2 Cholesky decomposition

Cholesky decomposition 39

$$[K] = [L][L]^T$$

$$\begin{bmatrix} l_{11} & L_{21}^T \\ 0 & L_{22}^T \end{bmatrix}$$

$$\begin{bmatrix} l_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} k_{11} & K_{12}^T \\ K_{21} & K_{22} \end{bmatrix}$$

$$l_{11} = \sqrt{k_{11}}$$

$$K_{21} = L_{21} l_{11} \rightarrow L_{21} = K_{21}/l_{11}$$

$$L_{22} L_{22}^T = K_{22} - \underbrace{L_{21} L_{21}^T}_{\text{known}}$$

Takes advantage of the symmetry, lower storage requirement  
~two times faster than LU decomposition

The previous method being costly, it is obvious to take advantage of the symmetry of  $K$ . The new method is summarized on the slide.

Figure 11.9

### 11.2.3 Frontal method

This consists in never assembling the stiffness matrix:

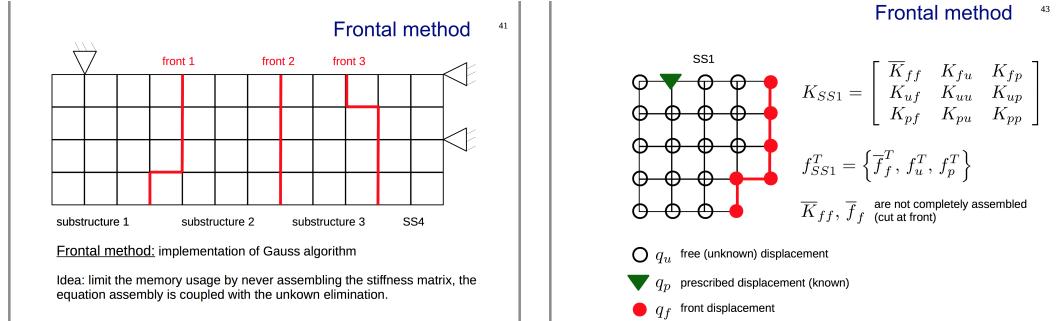


Figure 11.10

Figure 11.11

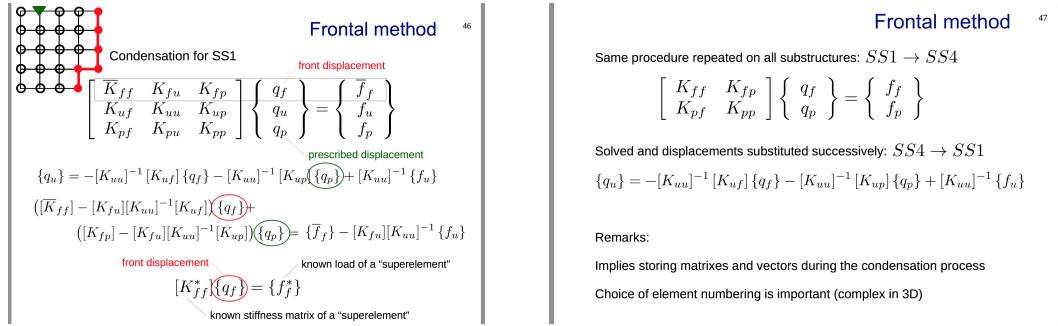


Figure 11.12

Figure 11.13

### 11.2.4 Iterative methods

The method consists in, starting from an initial guess, finding successive approximations. Pre-conditioners determine the efficiency of the method. The total computational work is not known as the iteration number is not known, but the work evolution is less high. Memory storage grows linearly, well adapted for very large scale problems. The problem is not solved exactly but we have a convergence.

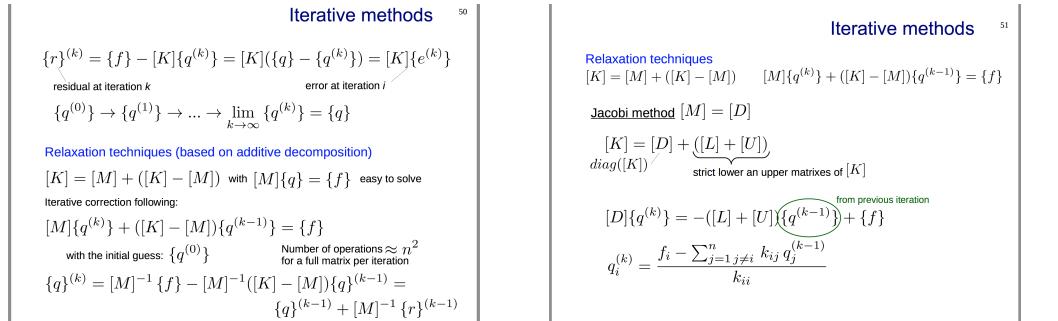


Figure 11.14

Figure 11.15

Iterative methods	
<b>Relaxation techniques</b>	52
$[K] = [M] + ([K] - [M]) \quad [M]\{q^{(k)}\} + ([K] - [M])\{q^{(k-1)}\} = \{f\}$ <p><b>Gauss-Seidel method</b> <math>[M] = [D] + [L]</math></p> $[K] = ([D] + [L]) + [U]$ $([D] + [L])\{q^{(k)}\} = -[U]\{q^{(k-1)}\} + \{f\}$ $q_i^{(k)} = \frac{f_i - \sum_{j=1}^{i-1} k_{ij} q_j^{(k)}}{k_{ii}} - \sum_{j=i+1}^n k_{ij} q_j^{(k-1)}$ <p style="text-align: center;">from current iteration    from previous iteration</p> <p><math>q_i^{(k)}</math> values of the current iteration depend less and less on the previous <math>q_j^{(k-1)}</math> (<math>j &gt; i</math>) than on the current values <math>q_j^{(k)}</math> (<math>j &lt; i</math>) – faster rate of convergence (and less storage requirement)</p>	

Figure 11.16

Iterative methods	
<b>Relaxation techniques</b>	53
The spectral radius (largest eigenvalue) of $(-[M]^{-1}([K] - [M]))$ determines the convergence behavior	
<b>Minimization techniques (conjugate gradients)</b>	
For positive definite $[K]$ : $\{q\}$ solves $[K]\{q\} = \{f\}$ and minimizes $f(\{q\}) = \frac{1}{2}\{q\}^T [K]\{q\} - \{f\}^T \{q\}$	
<b>Idea:</b> search for the minimum of $f$	
$\alpha_k = \frac{\{r\}^T r}{r^T [K] r}$ method of the steepest descent = looking for the best improvement in the direction of the negative gradient	
$\{q\}^{(k+1)} = \{q\}^{(k)} + \alpha_k \{r\}^{(k)}$	
$\{r\}^{(k+1)} = \{r\}^{(k)} - \alpha_k [K]\{r\}^{(k)}$	
Crucial quantity for success: $\kappa = \frac{\lambda_{max}([K])}{\lambda_{min}([K])}$ spectral condition number	

Figure 11.17

Iterative methods	
<b>Preconditioners</b>	54
Drawback of the previous methods: convergence is slower for more accurate discretization.	
The convergence rate is linked to $\lambda_{max}/\lambda_{min}$ therefore it can be accelerated by a "preconditioning" of the system according to	
$[K]\{q\} = \{f\} \rightarrow [M]^{-1}[K]\{q\} = [M]^{-1}\{f\}$	
matrix $[M]^{-1}$ should be as close to $[K]^{-1}$ as possible, but not too expensive to be computed!	
Simplest (and useless) choice: $[M] = [I]$	
Best (and most expensive) choice: $[M] = [K]$	
Example: Jacobi preconditioner $[M] = [D]$	

Figure 11.18

# Chapter 12

## Symmetry, repeatability, modularity

### 12.1 Notion of symmetry

To have symmetry, one must have the symmetry of the **geometry, material properties** and **the boundary conditions**. The two main symmetries are **plane** and **axial** symmetry.

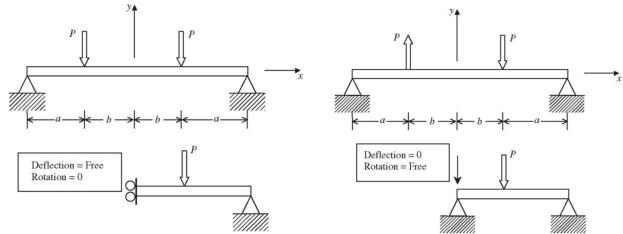
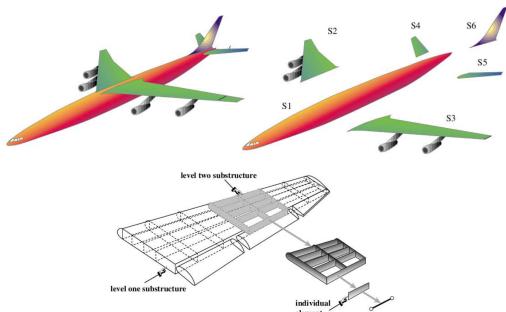


Figure 12.1

We have to take care when using a plane symmetry because some degrees of freedom have to be blocked as seen in the example. Anti symmetry can also be exploited. Axial symmetry is also used for example for a cooling tower, where the whole geometry can be retrieved by revolution of a 2D section (most turbomachinery have sectorial symmetry).

This allows to considerably decrease the computational time because most of the times the 3D problem becomes a 2D problem. Be aware that this is only applicable when the three conditions are met!

### 12.2 Notion of repeatability and modularity



consists in a sub-structure within the global structure (S1 to S6 in the figure). Three main advantages:

Modular structures are more and more appreciated due to the lower cost. Complex structures can be decomposed into simpler parts as depicted in the picture. For sure the connection of the boundaries has to be performed to transmit the forces, for this we use **super-elements**. A superelement is a grouping of finite elements that usually

- facilitate the division of the task in the engineering department of a company;
- advantage of symmetry, modularity and repeatability;
- reduce the computational effort.

The degrees of freedom in such elements can be divided into internal degrees of freedom which are not connected to the DOF of the other elements, and the boundary DOF which lie on the frontier between superelements. The stiffness matrix is written as:

$$\begin{bmatrix} K_{bb} & K_{bi} \\ K_{ib} & K_{ii} \end{bmatrix} \begin{bmatrix} q_b \\ q_i \end{bmatrix} = \begin{bmatrix} f_b \\ f_i \end{bmatrix} \quad (12.1)$$

where the indices b and i means boundary and internal. The second equation gives:

$$K_{ib}q_b + K_{ii}q_i = f_i \quad \Rightarrow q_i = K_{ii}^{-1}f_i - K_{ii}^{-1}K_{ib}q_b, \quad (12.2)$$

provided  $K_{ii}$  is invertible. This injected in first equation:

$$K_{bb}q_b + K_{bi}K_{ii}^{-1}f_i - K_{bi}K_{ii}^{-1}K_{ib}q_b = f_b \quad \Rightarrow \underbrace{(K_{bb} - K_{bi}K_{ii}^{-1}K_{ib})}_{\bar{K}_{bb}} q_b = \underbrace{f_b - K_{bi}K_{ii}^{-1}f_i}_{\bar{f}_b}. \quad (12.3)$$

This procedure is called **condensation of the stiffness equations**, allowed only in the condition  $K_{ii}$  not singular (if all rigid body motion are blocked). The condensed superelement can be seen as an individual elemeent described by (12.3). Then we proceed to assembly and the internal DOF can be retrieved by going back to each superelement.

# Chapter 13

## Verification & Validation

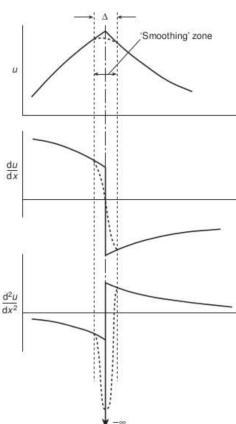
Keep in mind that each level of modeling introduces its own error in the process. But how can we be sure of the relevance of the finite element model.

- the verification process consists in determining the reliability of the computational model wrt the real world problem. Here we will focus on comparing the theoretical values and not the real world;
- validation is the assessment of the accuracy of the numerical results with respect to experimental data. This involves **calculation verification** comparing the discrete solution with the exact one of the mathematical problem, and **code verification** that consists in checking the difference between the mathematical solution and its algorithm implementation.

### 13.1 Criteria for convergence

There are 3 criteria to ensure convergence when the mesh size tends to 0:

- **Criterion 1:** the displacement function has to be such that it does not permit the deformation of an element when rigid body motion. The violation of this criterion induces no ability to get exact displacements when size tending to 0.
- **Criterion 2:** the displacement function has to be such that if the nodes are compatible with constant strain condition, such constant strains will be obtained. When the element get smaller, there is constant strain within and has to be able to represent it.
- the displacement functions have to be such that the strains at the interface of elements are finite.



Let's look to the third condition. As the stresses and strains only involves first order derivatives, the displacement function can be approximated by a linear approximation. In the figure is investigated the continuity between two elements. When discontinuity in  $u$ , the first-order derivative is finite but the second one not. Criterion 3 can be restated as when applying the energy equations to the whole structure, discontinuity in the displacements between two element cannot occur. When violation an artificial contribution stating the work of strain and stresses has to be added between elements. Element violating one of the criterion is called **non-conforming element**.

Figure 13.1

## 13.2 The patch test for solid elements

The above criteria can be checked by means of a patch test. A patch is the set of all the elements attached to a given node. The idea is that a good element has to solve simple problems exactly:

- **Displacement Patch Test (DPT):** the patch has to reproduce exactly rigid body modes and constant strain states when the corresponding boundary displacements are applied to the patch.
- **Force Patch Test (FPT):** the patch must reproduce exactly constant stresses when the corresponding boundary forces are applied.

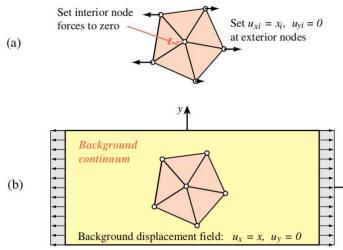


Figure 13.2

Consider the DPT, the idea is that for infinitely small mesh, the strain is almost constant in an element and thus the displacement distribution is linear. At the limit two cases should be considered: the rigid body mode and the constant strain states. For figure (a) we apply  $u = x, v = 0$  and thus  $\epsilon_x = \partial u / \partial x = 1$ . The verification is:

- evaluate the displacement on the external nodes and apply them as prescribed displacement;
- forces at internal DOF set to 0;
- solve  $Kq = f$  for interior nodes.
- compute the strain field over the elements, components vanish except  $\epsilon_x = 1$  at any point.

Similar procedure for FPT.

## 13.3 A short glimpse into error estimation techniques

The farness/closeness of the finite element displacement  $u^h$  wrt the theoretical  $u$  is measured through distance, norm in math. The energy norm is defined as:

$$\|e\|_W = \sqrt{\int_V (\epsilon - \epsilon^h)^T (\tau - \tau^h) dV} = \sqrt{\int_V (\epsilon - \epsilon^h)^T H(\epsilon - \epsilon^h) dV}. \quad (13.1)$$

Alternatively  $L_2$  norms can be defined for displacement, strain and stresses:

$$\begin{cases} \|e_u\|_{L_2} = \sqrt{\int_V (u - u^h)^T (u - u^h) dV} \\ \|e_\epsilon\|_{L_2} = \sqrt{\int_V (\epsilon - \epsilon^h)^T (\epsilon - \epsilon^h) dV} \\ \|e_\tau\|_{L_2} = \sqrt{\int_V (\tau - \tau^h)^T (\tau - \tau^h) dV} \end{cases} \quad (13.2)$$

A relative error can be defined as for the energy norm:

$$\eta = \frac{\|e\|_W}{\sqrt{\int_V \epsilon^T \tau dV}}. \quad (13.3)$$

Only two main approaches allow for enhancing the computational procedure:

- the h-element technique: decreasing the mesh size, or adapting it;
- the p-element technique: improving the finite elements by:

1. increasing the number of nodes while keeping the same number of DOF (Lagrange);
2. increasing the number of DOF per node (Hermite);
3. adopting a mixed approach.

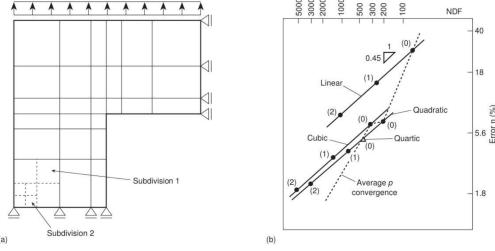


Figure 13.3

These methods are depicted on the figure. Convergence rate for a 2D structure. NDF ( $= Q$ ) is the number of degrees of freedom. (0), (1), and (2) refer to the initial mesh, the refined mesh with subdivision 1, and the refined mesh with subdivision 2, respectively. The different curves refer to the order of the polynomial used for the shape functions. The predicted optimal order is:

$$\|e\|_W \rightarrow O(h^p), \quad (13.4)$$

where  $h$  is the size of the element and  $p$  the order of the polynomial. In 2D the number of DOF  $Q$  can be used:

$$\|e\|_W \rightarrow O(Q^{-p/2}). \quad (13.5)$$

Two final conclusions:

- singularities (like the sharp corner) degrade the actual convergence with respect to the theoretical order of convergence. For smoother problems, the error curves can be closer to (but still always larger than) the theoretical results;
- an increase of the polynomial order is not always advisable. Second-order polynomials usually constitute a good compromise between accuracy, number of degrees of freedom, and complexity of the shape functions.

In practice the order of the polynomial is determined by the element used, the results can be improved by mesh adaptivity techniques:

1. **element subdivision:** elements are divided into smaller ones;
2. **remeshing:** new mesh is created, computationally more expensive but best results;
3. r-refinement: the mesh connectivity is unchanged but the nodes coordinate change (rarely used in practice because error impossible to control).