Finite element method isoparametric

Abstract :

This document presents the bases of the finite elements isoparametric introduced into *Code_Aster* for the modelization of the continuums 2D and 3D. One first of all recalls the transition of a strong formulation to a variational formulation, then one details the discretization by finite elements: use of an element of reference, computation of the shape functions and evaluating of the elementary terms. One also briefly describes the principle of the assembly of these terms and the imposition of the boundary conditions, and one evokes the methods of matric resolution used. Finally are exposed the main steps of a computation by finite elements such as it are conceived and established in *Code_Aster*.

Contents

1 Introduction

the finite element method is employed in many scientific disciplines to solve partial derivative equations. It makes it possible to build a simple approximation of the unknowns to transform these continuous equations into a system of equations of finished size, which one can schematically write in the following form:

$$[A].\{U\}=\{L\} \quad (1)$$

where $\{U\}$ is the vector of the unknowns, $[A]$ a matrix and $\{L\}$ a vector.

Initially, one transforms the partial derivative equations into an integral formulation (or strong ***formulation*** of the problem), often this first integral form is modified (weakened) by means of the formula of Green (one obtains a weak formulation ***then***). The approximate solution is sought like linear combination of functions given. These functions must be simple but enough general to be able "well" to approach the solution. They must in particular make it possible to generate a space of finished size which is as close as one wants space of functions in which the solution is. From this old idea (method of the balanced residues), the various ways of choosing these functions cause various numerical methods (collocation, methods spectral, finite elements, etc).

The originality of the finite element method is to take as functions of approximation of the polynomials which are null on almost all the field, and thus take part in computation only in the vicinity of a particular point. Thus, the matrix $[A]$ is very hollow, containing only the terms of interaction between "close points", which reduces the computing time and the core memory necessary to storage. Moreover, the matrix $[A]$ and the vector $\{L\}$ can be built by assembly of matrixes and elementary vectors, calculated locally.

2 Obtaining a variational formulation

One can obtain the variational formulation of a problem starting from the partial derivative equations, by multiplying those by functions tests and while integrating by parts. In mechanics of solids, the weak formulation then obtained is identical to that given by the Principle of the Virtual wors and in the conservative case, the minimization of the total potential energy of structure. Let us note however that for certain problems, the equations of the model are easier to establish in the variational frame (case of the plates and the shells for example).

2.1 Modelization of the physical problem – Principles and notation

a physical system is generally modelled by partial derivative equations which act on unknowns $\mathbf{u}$ who can be:
•A scalar like the temperature in the problems of thermal;
•A vector like displacements in the problems of mechanics;
•A tensor like the stresses in the problems of mechanics;

One can also use several fields of unknowns simultaneously, connected by partial derivative equations. They are coupled *problems*. In Code_Aster, one can quote as example the problems of thermo-hydro-mechanics which couple displacements, pressure and temperature.
The fields of unknowns are parameterized by:
•The space, which can be described by a coordinate system Cartesian or any other type of parameterization. In the continuation of the document, one will note it x ;
•Time, noted t ;

2.2 Equations of the system

a continuous physical system can be represented by *a system of equations* with the partial derivatives which one will write in the field Ω :

$$\mathbf{L}(\mathbf{u})+\mathbf{f}=\mathbf{0} \quad \text{dans} \quad \Omega \quad (2$$

This system is associated with the boundary conditions on the border Γ of the field Ω :

$$\mathbf{C}(\mathbf{u})=\mathbf{h} \quad \text{sur} \quad \Gamma=\partial\Omega \quad (3)$$

the differential operator can express himself on several partial derivative equations. One could write:

$$\begin{aligned} L_1(\mathbf{u})+f_1&=0 \\ L_2(\mathbf{u})+f_2&=0 \\ &\dots \end{aligned} \tag{4}$$

$L_i(\mathbf{u})$ is a differential operator acting on the vector of the unknowns $\mathbf{u}$. In a more general way, the differential operator $L_i(\mathbf{u})$ is written according to the unknowns and their partial derivatives:

$$L_i\left(\mathbf{u},\frac{\partial \mathbf{u}}{\partial x_1},\dots,\frac{\partial^2 \mathbf{u}}{\partial x_1.\partial x_2},\dots,\frac{\partial^m \mathbf{u}}{\partial x_\alpha^m},t,\frac{\partial \mathbf{u}}{\partial t},\dots,\frac{\partial^p \mathbf{u}}{\partial t^p},\right) \tag{5}$$

Such an operator is known as of order m in space and order p in time. If it does not depend on time (and its derivatives), it is said that the problem is steady. In the continuation of the document one will consider only the steady problems.

2.3 Method of the balanced residues – strong integral Formulation

One will define the residue $\mathbf{R}(\mathbf{u})$ as being the quantity cancelling itself when $\mathbf{u}$ is the solution of the physical problem:

$$\mathbf{R}(\mathbf{u})=\mathbf{L}(\mathbf{u})-\mathbf{f}=\mathbf{0} \quad \text{dans} \quad \Omega \tag{6}$$

the method of the balanced residues consists:
1 A to build a solution approached $\mathbf{u}$ by the linear combination of judiciously selected functions

$$\mathbf{u}(\boldsymbol{x})=\sum_{i=1}^{N} c_i.\boldsymbol{\phi}_i(\boldsymbol{x}) \tag{7}$$

Where $\boldsymbol{\phi}_i(\boldsymbol{x})$ are the shape functions of the approximation and c_i the coefficients to be identified.
2 A to solve the system in integral form:

$$\begin{aligned} &\text{Trouver} \quad \mathbf{u}\in E_u \quad \text{tel que} \quad \forall \mathbf{P}\in E_P \\ &\text{Avec} \quad W=\int_\Omega \mathbf{R}(\mathbf{u}).\mathbf{P}(\mathbf{u}).d\Omega+\int_\Gamma [\mathbf{C}(\mathbf{u})-\mathbf{h}].\mathbf{P}(\mathbf{u}).d\Omega=0 \end{aligned} \tag{8}$$

We used the same weight functions for the principal system and the limiting conditions, but it is not compulsory. $\mathbf{P}(\mathbf{u})$ are the weight functions belonging to a set of functions E_P. The solution $\mathbf{u}$ belongs to the space E_u of the regular functions "sufficiently" (differentiable until the order *m*).
The choice of the weight functions $\mathbf{P}(\mathbf{u})$ makes it possible to create several methods:
•If the function $\mathbf{P}(\mathbf{u})$ is a distribution of Dirac, one obtains the collocation method by points.
•If the function $\mathbf{P}(\mathbf{u})$ is constant on subdomains, one obtains the collocation method by subdomains.
•If the weight functions $\mathbf{P}(\mathbf{u})$ use the same shape functions $\boldsymbol{\phi}_i(\boldsymbol{x})$ as the approximation of the solution (7), one obtains the method of Galerkin.
The strong integral ***form thus is obtained***.

2.4 Weak integral formulation

the integral formulation (8) requires differentiable spaces of function to the order m for E_u. The weak formulation consists in carrying out an integration by parts (by application of the formula of Green) of the system (8). On the other hand one increases the requirements for regularity on the weight functions $\mathbf{P}(\mathbf{u})$. The formula of Green is stated as follows:

$$\int_\Omega \mathbf{u}.\nabla.\mathbf{P}.d\Omega=-\int_\Omega \mathbf{P}.\nabla.\mathbf{u}.d\Omega+\int_\Gamma \mathbf{u}.\mathbf{P}.\mathbf{n}.d\Gamma \tag{9}$$

where $\boldsymbol{n}$ is the outgoing norm at the border Γ of the field.

3 Finite element method

3.1 general Principles

the search of a suitable approximate function on all the field becomes difficult in the general case of a geometry of an unspecified form. The idea of the finite element method is thus to build this approximation in two times:

•To identify subdomains Ω_e geometrically simple which pave the field;

•To define a function approached on each subdomain;

A certain number of characteristics of this construction are thus had a presentiment of:

•The paving of the field Ω_e by the subdomains Ω_e must be as precise as possible;

•The function approached on the subdomain must observe conditions of continuity between the various subdomains;

•The function approached on the subdomain must have coherent properties with the conditions of derivability and in keeping with the physical description of the solution (what can imply to use a weakened formulation for example).

3.2 Approximation of the geometry

3.2.1 Principle

One identifies N_e the subdomains (or *elements*) Ω_e which pave the space Ω of solid:

$$\Omega = \sum_{e=1}^{N_e} \Omega_e \tag{10}$$

Let us note $x_{\alpha=1,3}$ the punctual coordinates $\boldsymbol{x}$ in the absolute coordinate system. The geometry of the subdomain is built with a nodal approximation, that is to say for *an element* with N_{nd} nodes:

$$\boldsymbol{x}^e = \sum_{i=1}^{N_{nd}} \boldsymbol{x}_i^e . \bar{\boldsymbol{N}}_i^e \text{ or } x_\alpha^e = \sum_{i=1}^{N_{nd}} x_{\alpha,i}^e . \bar{N}_i^e \tag{11}$$

This paving (mesh) is an operation being able to be complex, especially in 3D. There exist general algorithms to net. One uses triangles or quadrangles in 2D and tetrahedrons or hexahedrons in 3D (more some elements being used as connections). The triangles and tetrahedrons what is called give free meshes , the quadrangles and the hexahedrons form structured meshes . The free meshes are relatively easy to build thanks to largely tested techniques: cells of Voronoï building a triangulation of Delaunay or methods of propagation (methods known as frontal), the structured meshes are much more delicate to generate. The mesh necessarily induced a geometrical error of discretization For example, on the figure (1), one sees that a curved border only is imperfectly approached by linear elements.

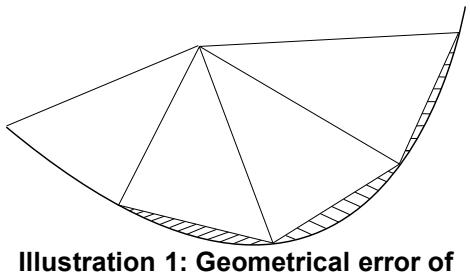
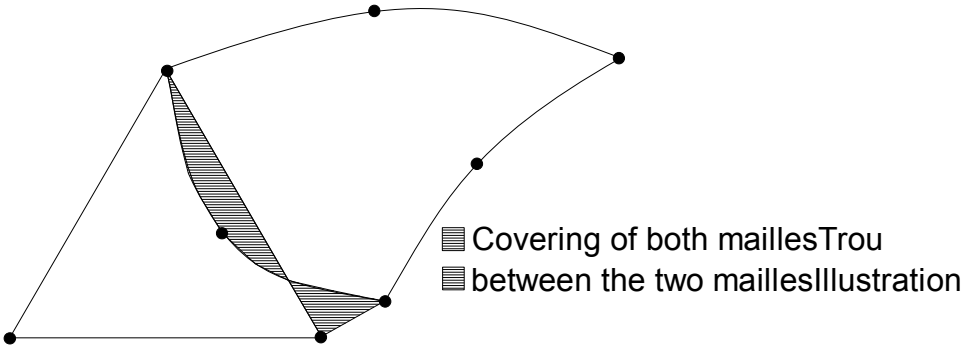


Illustration 1: Geometrical error of discretization

In the same way the mesh must be in conformity: no holes or of covering (see figure (2).



2: Nonconformity of the mesh

to observe this condition of *conformity*, it is enough to two rules:

1.Each element must be defined in a single way starting from the coordinates of its geometrical nodes (and not those of its neighbors!);

2.The border of an element must be defined in a single way starting from the nodes of these borders, these nodes being common between the elements dividing this border.

These conditions of conformity are an important difference compared to finished volumes which do not have these requirements. The paving of the field makes it possible to apply the finite element method to complex geometries, contrary to the methods by finite differences. The geometrical paving of the field induces a first *error*: it is not possible, in the general case, to represent a real geometry by a mesh by regular polygons, in particular on the border of the field.

"***A beautiful mesh is a good mesh"***

3.2.2 Elements of reference

The computation of the shape functions for an unspecified element can be rather complicated. This is why one often prefers to bring back oneself to an element known as of reference, from which one can generate all the elements of the same family by a geometrical transformation. The shape functions are then calculated on this noted generic element Ω_r, and the transport of the quantities on the real element Ω_e is accomplished thanks to the knowledge of the geometrical transformation.

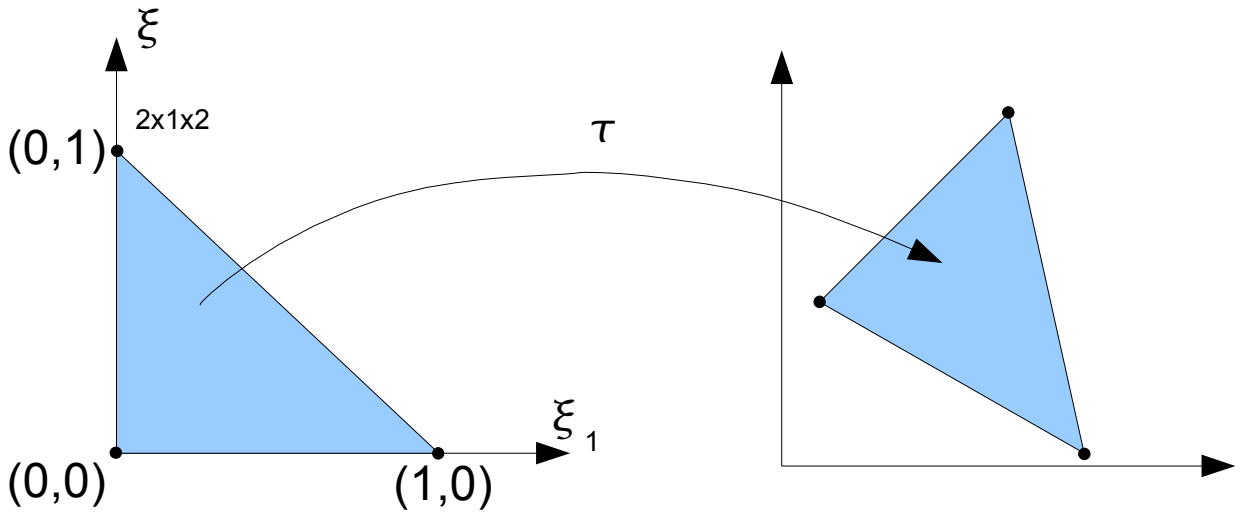


Illustration 3: Transition of the space of reference to real space

the points of the element of reference will be described in parametric terms of coordinates $\xi_{\alpha=1,3}$. The transformation τ must be bijective and transform the tops and sides of the element of reference into tops and sides of the real element:

$$\xi_\alpha \overset{\tau}{\rightarrow} x_\alpha \qquad (12)$$

3.2.3 Interpolation functions geometrical

the geometry of the element is thus approximate by the means of interpolation functions geometrical. These noted functions $\bar{N}(\boldsymbol{\xi})$ are defined on the element of reference; they make it possible to know the unspecified x_α punctual coordinates of the real element from its coordinates ξ_α of its antecedent in the element of reference and of the coordinates x_α^i of the nodes (of local number I) of the real element:

$$\boldsymbol{x}^e=\sum_{i=1}^{N_{nd}} \boldsymbol{x}_i^e . \bar{\boldsymbol{N}}_i^e \text{ or } x_\alpha^e=\sum_{i=1}^{N_{nd}} x_{\alpha,i}^e . \bar{N}_i^e \quad (13)$$

3.2.4 Jacobian matrix of the transformation

the jacobienne of the transformation is the matrix of derivatives partial of the real coordinates x_α compared to the coordinates ξ_α in the element of reference:

$$J_{\alpha\beta}=\frac{\partial x_\alpha}{\partial \xi_\beta} \quad (14)$$

By taking account of the definition of the coordinates x_α according to the coordinates $x_{\alpha,i}$ of the nodes, one obtains an equivalent statement of the jacobian matrix:

$$J_{\alpha\beta}=\sum_{i=1}^{N_{nd}} \frac{\partial \bar{N}_i}{\partial \xi_\beta} . x_{\alpha,i} \quad (15)$$

Where $\frac{\partial \bar{N}_i}{\partial \xi_\beta}$ are the terms of the matrix $\left[\frac{\partial \bar{\boldsymbol{N}}}{\partial \boldsymbol{\xi}}\right]^T$, of which the number of lines is the number of directions of space, and the number of columns the number of nodes of the element. Let us note that the matrix $\left[\frac{\partial \bar{\boldsymbol{N}}}{\partial \boldsymbol{\xi}}\right]^T$ depends only on the definition of the element of reference and not of that of the real element. The determinant of the jacobian matrix, useful in computations which will follow, is called the jacobian of the geometrical transformation. It is non-zero when the transformation τ which makes pass from the element of reference to the real element is bijective, and positive when τ respects the directional sense of space.

$$J=\det\left[\frac{\partial \bar{\boldsymbol{N}}}{\partial \boldsymbol{\xi}}\right]\geq 0 \quad (16)$$

3.3 Representation of the unknowns

to solve the problem, one considers an approximation by finite elements of an unknown field. Spaces E_P and E_u are represented by spaces discrete E^h. There are two equivalent ways to represent the unknowns in an element: by the coefficients of their polynomial approximation, or by their nodal values. These two possibilities correspond to the two ways complementary to define an element: by the data of a base of students' rag processions, or by the data of the shape functions associated with the nodes. In a general way, one builds the function approached by writing the following linear relation on each element:

$$\mathbf{u}^e(\boldsymbol{\xi})=\sum_{i=1}^{N_{nd}} a_i^e . \boldsymbol{\phi}_i^e(\boldsymbol{\xi}) \quad (17)$$

Where are $\boldsymbol{\phi}_i^e(\boldsymbol{\xi})$ to them independent linear functions. They constitute ***the base*** of the approximation, the general parameters of the approximation being the coefficients a_i.

3.3.1 Nodal approximation

the first idea of the finite element method is to build approximation of a nodal type for which the coefficients $u_i=a_i$ correspond to the solution in these nodes:

$$\mathbf{u}^e(\boldsymbol{\xi}) = \sum_{i=1}^{N_{nd}} u_i^e . \boldsymbol{N}_i^e(\boldsymbol{\xi}) \tag{18}$$

One then obtains a nodal approximation with $\boldsymbol{N}_i^e(\boldsymbol{\xi})$ the interpolation functions on the element of reference. On each one of these subdomains one builds an approximate function different from one subdomain to another. The approximation finite elements is *elementary* because the function depends only on the nodal values constituting the element:

$$\mathbf{u}^e(\boldsymbol{x}) = \sum_{i=1}^{N_{nd}} u_i^e . \boldsymbol{N}_i^e(\boldsymbol{x}) \tag{19}$$

an element is *isoparametric* when it is based on identical interpolations for its geometry and its unknowns: $\bar{N}(\boldsymbol{\xi}) = N(\boldsymbol{\xi})$.
To ensure the continuity of the solution on the element and, possibly, the continuity of its derivatives, one needs that the functions $\boldsymbol{N}_i^e(\boldsymbol{\xi})$ are continuous and, possibly, with continuous derivatives.
In the same way if one wants to ensure the continuity of the solution and of its derivatives at the borders of the elements (conformity of the approximation), it is necessary that the solution and its derivatives depend in a single way of the nodal variables on the nodes of the border.

3.3.2 Base polynomial

the way simplest to define an element is to choose a polynomial base made up of a certain number of independent students' rag processions. For a given unknown, the number of students' rag processions used must be equal to the number of nodal variables, i.e. with the number of nodes used to represent the unknown. One generally defines the polynomial base on the element of reference; it contains students' rag processions of the form $\xi_1^{\gamma} . \xi_2^{\delta} . \xi_3^{\varepsilon}$, where γ , δ and ε are positive or null whole exhibitors. The degree of such a students' rag procession is the integer $\gamma+\delta+\varepsilon$. The base is known as complete of degree n when all the students' rag processions of degree n are present. In certain cases, incomplete bases are employed. One notes $P_p(\boldsymbol{\xi})$ $p^{ième}$ the students' rag procession of the base (which understands some m). The components of the vector displacement $\mathbf{u}(\boldsymbol{\xi})$ in the element are then given by the formula:

$$u_\alpha(\boldsymbol{\xi}) = \sum_{p=1}^{m} a_{\alpha,p} . P_p(\boldsymbol{\xi}) \tag{20}$$

One will note $\boldsymbol{\Pi}$ the matrix giving the values taken by the students' rag processions of the polynomial base on the nodes of the element of reference:

$$\Pi_{Ip} = P_p(\boldsymbol{\xi}_I) \tag{21}$$

where p is the sequence number of the students' rag procession in the base, I number of the node locally to the element and the $\boldsymbol{\xi}_I$ coordinated of the node I in the element of reference. This matrix is square, its dimension is the square amongst nodes of the element.
With the node I displacement u_α^I is worth:

$$u_{I,\alpha} = a_{\alpha,p} . \Pi_{Ip} \tag{22}$$

One distinguishes three large element types finished frequently used:
•finite elements of Lagrange which rest on bases polynomial complete and different standard from geometries (symplectic for the triangles and the tetrahedrons, with tensorial structure for the quadrangles and the hexahedrons or of prismatic type);
•the finite elements of the Serendip type, which are of the finite elements of Lagrange with incomplete bases;
•the finite elements of Hermit, of utmost precision, which use the nodal unknowns and their derivatives;

Finite elements of Lagrange symplectic

to determine if a polynomial base is complete with the elements symplectic, it is enough to use the triangle of Pascal:

Linear

Constant order			1		
		ξ_1		ξ_2	
Quadratic formula		$(\xi_1)^2$	$\xi_1.\xi_2$	$(\xi_2)^2$	
	$(\xi_1)^3$	$(\xi_1)^2.\xi_2$	$\xi_1.(\xi_2)^2$	$(\xi_2)^3$	
Order 4	$(\xi_1)^4$	$(\xi_1)^3.\xi_2$	$(\xi_1)^2.(\xi_2)^2$	$\xi_1.(\xi_2)^3$	$(\xi_2)^4$

a complete polynomial base of order two comprises six students' rag processions: $\{1;\xi_1;\xi_2;\xi_1^2;\xi_2^2;\xi_1.\xi_2\}$ and thus the geometric standard support will be a triangle with six nodes.

Finite elements of Lagrange with tensorial structure

to describe of the finite elements quadrangular (or hexahedral), it is enough to take complete polynomials of the order given and to make the product of it.

Quadratic	Constant	order	Linear	Cubic
Constant	1	ξ_1	$(\xi_1)^2$	$(\xi_1)^3$
Linear	ξ_2	$\xi_1.\xi_2$	$(\xi_1)^2.\xi_2$	$(\xi_1)^3.\xi_2$
Quadratic	$(\xi_2)^2$	$\xi_1.(\xi_2)^2$	$(\xi_1)^2.(\xi_2)^2$	$(\xi_1)^3.(\xi_2)^2$
Cubic	$(\xi_2)^3$	$\xi_1.(\xi_2)^3$	$(\xi_1)^2.(\xi_2)^3$	$(\xi_1)^3.(\xi_2)^3$

a polynomial base “on” - complete of order two for a quadrangular element comprises nine students' rag processions: $\{1;\xi_1;\xi_1^2;\xi_2;\xi_2^2;\xi_1.\xi_2;\xi_1.\xi_2^2;\xi_1^2.\xi_2;\xi_1^2.\xi_2^2\}$, which means nine nodes. Such an element comprises terms of order 3 and 4.

Finite elements of Serendip

the elements of Serendip, for a polynomial of order s, exclude the cross terms from degree higher than $s+1$ not to have nodes inside the elements. For example, for an element of Serendip of order two, the students' rag processions will be $\{1;\xi_1;\xi_1^2;\xi_2;\xi_2^2;\xi_1.\xi_2;\xi_1.\xi_2^2;\xi_1^2.\xi_2\}$, that is to say eight nodes.

3.3.3 Shape functions

an equivalent way to define a finite element is to give, for each unknown, the statement of the shape functions of the element. For a given scalar unknown (component of displacement according to there for example), there is as much as nodes where the unknown must be calculated. In much of case, one uses the same shape functions for all the components of an unknown vector, but it is not compulsory. In what follows, it will be supposed however to simplify the writings that it is the case.

The shape functions can be defined on the real element Ω_e : they then are noted $N^e(\mathbf{x})$, they depend on the geometry of the real element, and are thus different from one element to another. It is simpler to express them on the element of reference, which gives the functions $N(\boldsymbol{\xi})$ independent of the geometry of the real element. Let us recall that these functions are polynomial on the element, and that the shape function associated with a node given there takes the value one, whereas it is cancelled in all the other nodes of the element. The unknowns are described then like linear combination of the shape functions, the coefficients $u_{\alpha,i}$ of the combination being called the nodal variables:

$$u_\alpha(\boldsymbol{\xi})=\sum_{i=1}^{N_{nd}} u_{\alpha,i}.N_i(\boldsymbol{\xi}) \tag{23}$$

By means of the transformation τ between the element of reference and the real element:

$$\xi_\alpha \overset{\tau}{\rightarrow} x_\alpha \tag{24}$$

There a:

$$u_\alpha(\boldsymbol{\xi}) = \sum_{i=1}^{N_{nd}} u_{\alpha,i} . N_i\left(\tau^{-1}(\boldsymbol{x})\right) \tag{25}$$

3.3.4 Correspondence between polynomial base and shape functions

One are two relations. The first comes from the approximation of the solution by a polynomial base:

$$u_\alpha(\boldsymbol{\xi}) = \sum_{p=1}^{m} a_{\alpha,p} . P_p(\boldsymbol{\xi}) \tag{26}$$

the second is the nodal approximation:

$$u_\alpha(\boldsymbol{\xi}) = \sum_{i=1}^{N_{nd}} u_{\alpha,i} . N_i(\boldsymbol{\xi}) \tag{27}$$

the matrix giving the values taken by the students' rag processions of the polynomial base on the nodes of the element of reference:

$$\Pi_{Ip} = P_p(\boldsymbol{\xi}_I) \tag{28}$$

In a node I, one wrote the following polynomial approximation;

$$u_{I,\alpha} = a_{\alpha,p} . \Pi_{Ip} \tag{29}$$

By injecting the equation (29) in the nodal statement (27), one obtains:

$$u_\alpha(\boldsymbol{\xi}) = \sum_{i=1}^{N_{nd}} a_{\alpha,p} . \Pi_{Ip} . N_i(\boldsymbol{\xi}) \tag{30}$$

By comparison with the polynomial approximation (26), one from of deduced the following relation between the polynomial base and the shape functions:

$$\Pi_{Ip} . N_i(\boldsymbol{\xi}) = P_p(\boldsymbol{\xi}) \tag{31}$$

In practice, one will find in the literature the writings of the nodal shape functions for the most current elements, according to the choice of the polynomial base.

3.4 Results of existence and unicity

One can write the problem in a more abstract way:

$$\begin{aligned} &\text{Trouver} \quad \mathbf{u} \in E_u \quad \text{tel que} \quad \forall \mathbf{v} \in E_v \\ &a(\mathbf{u},\mathbf{v}) = f(\mathbf{v}) \end{aligned} \tag{32}$$

E_u and E_v are vector spaces of functions defined on Ω. They are spaces of Hilbert.
$a(\mathbf{u},\mathbf{v})$ is a bilinear form on $E_u \times E_v$ (it was supposed that $\mathbf{L}(\mathbf{u})$ represents a linear physical problem compared to $\mathbf{u}$).
$f(\mathbf{v})$ is a linear and continuous form on E_v.
To establish the conditions of existence and unicity, one applies the theorem of Lax-Milgram. Initially, it is supposed that the solution belongs to the same space as the functions test $E_u = E_v$
If the form $a(\mathbf{u},\mathbf{v})$ is coercive i.e.:

$$\forall \mathbf{u} \in E_u \qquad a(\mathbf{u},\mathbf{u}) \geq c. \left\|\mathbf{u}\right\|_{E_u}^2 \qquad avec \ c > 0 \tag{33}$$

Then problem:

$$\begin{aligned} &\text{Trouver} \quad \mathbf{u} \in E_u \quad \text{tel que} \quad \forall \mathbf{v} \in E_u \\ &a(\mathbf{u},\mathbf{v}) = f(\mathbf{v}) \end{aligned} \tag{34}$$

admits one and only one solution.

4 Method of Ritz

the method of Galerkine, in certain cases, is equivalent making steady a functional calculus. It is the case if the bilinear form $a(\mathbf{u},\mathbf{v})$ is symmetric and positive:

$$\forall \mathbf{u},\mathbf{v}\in E_u \quad a(\mathbf{u},\mathbf{v})=a(\mathbf{v},\mathbf{u}) \quad \text{et} \quad a(\mathbf{u},\mathbf{u})\geq 0 \tag{35}$$

In this case the problem (34) admits one and only one solution $\mathbf{u}$ minimizes on E_u the following functional calculus:

$$\pi(\mathbf{u})=\frac{1}{2}.a(\mathbf{u},\mathbf{u})-f(\mathbf{u}) \tag{36}$$

From the mechanical point of view, that means that the principle of virtual power can be also written like the minimization of a scalar quantity: the total energy of structure. This way write the equilibrium is very frequently employed. We here will have some results of them.

We point out initially that a functional calculus is a function of a set of functions (and of its derivatives). This π functional calculus will be written. One will limit oneself to the formulations in displacement, knowing that there is the different one. In this case, the functional calculus π will be written:

$$\pi(\boldsymbol{u})=\pi\left(\boldsymbol{u},\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}\right) \tag{37}$$

For the conservative problems, one can show that to write that the first variation of π is null (condition of stationarity of the functional calculus) is equivalent applying the principle of the virtual wors, or to use the method of Galerkine by taking virtual displacements like weight function. One calls that the method of Galerkine consists starting from the problem with derivatives partial establishing the equilibrium of structure, that is to say:

$$\boldsymbol{L}(\boldsymbol{u})+\boldsymbol{f}=0 \text{ dans } \Omega \text{ avec } \boldsymbol{\sigma}.\boldsymbol{n}=\boldsymbol{g} \text{ sur } \Gamma_N \text{ et } \boldsymbol{u}=\boldsymbol{u}^D \text{ sur } \Gamma_D \tag{38}$$

One then seeks to solve the problem in integral form by means of weight functions which are of the same nature as the approximate solution:

$$\begin{gathered} W=\int_\Omega [\boldsymbol{L}(\boldsymbol{u})+\boldsymbol{f}].\boldsymbol{\psi}(\boldsymbol{u}).d\Omega=0 \\ \text{Avec } \boldsymbol{\sigma}.\boldsymbol{n}=\boldsymbol{g} \text{ sur } \Gamma_N \text{ et } \boldsymbol{u}=\boldsymbol{u}^D \text{ sur } \Gamma_D \end{gathered} \tag{39}$$

If one chooses like weight function the variation of the unknowns $\boldsymbol{\psi}=\delta\boldsymbol{u}$ and after having integrated by parts once, one obtains:

$$\delta\pi(\boldsymbol{u})=W(\boldsymbol{u})=0 \ avec \ \boldsymbol{u}=\boldsymbol{u}^D \ sur \ \Gamma_D \tag{40}$$

To find the form exact of the functional calculus is not immediate in the general case. In mechanics, for the conservative cases, it is that this functional calculus is equivalent to the total potential energy of the system. After discretization of the functional calculus (by an approximation finite elements), one finds oneself with a matric system strictly equivalent to that of the method of Galerkine (or its mechanical principle are equivalent, the method of the virtual powers).

Intuitively, it is understood that a weak variation $\delta\boldsymbol{u}$ of the solution is a field which can be kinematically admissible and which thus corresponds well to the assumptions of the method of the virtual powers.

5 Construction of the matric system

We now will present the various ingredients leading to the construction of the matric system which will make it possible to solve the problem.

5.1 New notation (notation of Voigt)

In order to understand well the construction of the discrete terms in the finite element method, we will use a more compact notation:

$\langle V \rangle$ is a vector line

$\{V\}$ is a vector column
$[A]$ is a matrix

Thus the geometrical interpolation is written according to three dimensions of space:

$$\begin{aligned} x^e = x_1^e = \langle x_{1,i}^e \rangle . \{N_i^e\} = \langle N_i^e \rangle . \{x_{1,i}^e\} \\ y^e = y_1^e = \langle y_{1,i}^e \rangle . \{N_i^e\} = \langle N_i^e \rangle . \{y_{1,i}^e\} \\ z^e = z_1^e = \langle z_{1,i}^e \rangle . \{N_i^e\} = \langle N_i^e \rangle . \{z_{1,i}^e\} \end{aligned} \quad (41)$$

Or in a more compact way in vectorial form:

$$\{\boldsymbol{x}^e\} = [N_i^e] . \{\boldsymbol{x}_i^e\} = \langle \boldsymbol{x}_i^e \rangle . [N_i^e]^T \quad (42)$$

With the matrix N of the shape functions. By considering an element with two nodes, one obtains in developed form:

$$\begin{Bmatrix} x^e \\ y^e \\ z^e \end{Bmatrix} = \begin{bmatrix} N_1^e & 0 & 0 & N_2^e & 0 & 0 \\ 0 & N_1^e & 0 & 0 & N_2^e & 0 \\ 0 & 0 & N_1^e & 0 & 0 & N_2^e \end{bmatrix} . \begin{Bmatrix} x_1^e \\ y_1^e \\ z_1^e \\ x_2^e \\ y_2^e \\ z_2^e \end{Bmatrix} \quad (43)$$

5.2 discretized System

One is placed in the case hyper elastic in small strains, the problem of mechanics to be solved one writes in a more compact way:

$$\begin{aligned} &\text{To find } \boldsymbol{u} \in E^h \text{ such as } \forall \tilde{\boldsymbol{u}} \in E^{\hat{h}} \\ &\text{with } a(\boldsymbol{u}, \tilde{\boldsymbol{u}}) + l(\tilde{\boldsymbol{u}}) = 0 \end{aligned} \quad (44)$$

With $a(\boldsymbol{u}, \tilde{\boldsymbol{u}})$ a bilinear, symmetric form which represents the potential energy of structure and $l(\tilde{\boldsymbol{u}})$ the potential[1] voluminal and surface forces:

$$\begin{aligned} a(\boldsymbol{u}, \tilde{\boldsymbol{u}}) &= \int_{\Omega^h} \boldsymbol{\varepsilon}(\tilde{\boldsymbol{u}}) : \boldsymbol{\sigma}(\boldsymbol{u}) . d\Omega^h \\ l(\tilde{\boldsymbol{u}}) &= \int_{\Omega^h} \boldsymbol{f} . \tilde{\boldsymbol{u}} . d\Omega^h + \int_{\Gamma_N^h} \boldsymbol{g} . \tilde{\boldsymbol{u}} . d\Gamma^h \end{aligned} \quad (45)$$

the discretization consists in choosing a base of space Ω^h and with calculating the terms of the matrix numerically $\boldsymbol{A}$ and vector $\boldsymbol{L}$. For that, one expresses the bilinear form $a(.,.)$ and the linear form $l(.)$ like a sum on elements, defined by basic field division:

$$\begin{cases} a(u_i, \tilde{u}_j) &= \sum_{\text{éléments } \Omega^e} \int_{\Omega^e} \sigma_{kl}(u_i) . \varepsilon_{kl}(\tilde{u}_j) . d\Omega^e \\ l(\tilde{u}_i) &= \sum_{\text{éléments } \Omega^e} \int_{\Omega^e} f_i . \tilde{u}_i . d\Omega^e + \int_{\Gamma_N^e} g_i . \tilde{u}_i . d\Gamma_N^e \end{cases} \quad (46)$$

the terms A_{ij}, which represent the interaction between two degrees of freedom i and j are built by *assembling* (the noted operation $\sum_{\text{éléments } \Omega^e} (\ldots)$) the contributions coming from each element which contains the corresponding nodes; one proceeds in the same way to build the second member vector L_i. These

1 potentiel1Le of the external forces does not depend on the displacement of structure, it is what is called a dead *loading* or *NON-follower*. In the cases of the large deformations, the loadings of type pressure cannot respect this assumption.

contributions, called elementary terms, are calculated during a loop on the elements and depend only on the only variables of the element Ω^e :

$$\begin{cases} a^e &= \int_{\Omega^e} \sigma_{kl}.\varepsilon_{kl}.d\Omega^e \\ l^e &= \int_{\Omega^e} f_i.w_i.d\Omega^e + \int_{\Gamma_N^e} g_i.w_i.d\Gamma_N^e \end{cases} \quad (47)$$

the relation between the tensor of the stresses of Cauchy $\boldsymbol{\sigma}$ and the displacements $\mathbf{u}$ is given by the behavior model, and is independent of the writing of the variational formulation. In the elastic case, one a:

$$\sigma_{ij}(w_i) = \Lambda_{ijkl}.\varepsilon_{ku}(w_i) \quad (48)$$

Λ_{ijkl} is the elasticity tensor of Hooke. This tensorial form is not very practical, one preferentially uses the notation of Voigt, which makes it possible to write:

$$\boldsymbol{\sigma} : \boldsymbol{\varepsilon} = \langle \sigma \rangle . \{\varepsilon\} \quad (49)$$

In Cartesian coordinates, one a:

$$\langle \sigma \rangle = \left\langle \sigma_{xx} \quad \sigma_{yy} \quad \sigma_{zz} \quad \sigma_{xy} \quad \sigma_{xz} \quad \sigma_{yz} \right\rangle \quad (50)$$

And a form modified of the components of strain to allow to express the contracted product, are:

$$\langle \varepsilon \rangle = \left\langle \varepsilon_{xx} \quad \varepsilon_{yy} \quad \varepsilon_{zz} \quad 2.\varepsilon_{xy} \quad 2.\varepsilon_{xz} \quad 2.\varepsilon_{yz} \right\rangle \quad (51)$$

important Remark:

In the integration of the constitutive laws, the components of shears of the stresses and strains used by Code_Aster *are:*

$$\langle \sigma \rangle = \left\langle \sigma_{xx} \quad \sigma_{yy} \quad \sigma_{zz} \quad \sqrt{2}.\sigma_{xy} \quad \sqrt{2}.\sigma_{xz} \quad \sqrt{2}.\sigma_{yz} \right\rangle$$
$$\langle \varepsilon \rangle = \left\langle \varepsilon_{xx} \quad \varepsilon_{yy} \quad \varepsilon_{zz} \quad \sqrt{2}.\varepsilon_{xy} \quad \sqrt{2}.\varepsilon_{xz} \quad \sqrt{2}.\varepsilon_{yz} \right\rangle$$

the product of these two vectors gives the same one well result as the double contracted product (49).

With this new notation, we have in elasticity:

$$\{\sigma\} = [A].\{\varepsilon\} \quad (52)$$

We set out again of the writing EF of the field of displacements:

$$\left\{\boldsymbol{u}^e\right\} = \left[N_i^e\right].\left\{\boldsymbol{u}_i^e\right\} = \left\langle \boldsymbol{u}_i^e \right\rangle.\left[N_i^e\right]^T \quad (53)$$

And, of similar way, the field of virtual displacements:

$$\left\{\tilde{\boldsymbol{u}}^e\right\} = \left[N_i^e\right].\left\{\tilde{\boldsymbol{u}}_i^e\right\} = \left\langle \tilde{\boldsymbol{u}}_i^e \right\rangle.\left[N_i^e\right]^T \quad (54)$$

By preoccupation with a simplification of the notations, one will omit the reference to the element. It is necessary of xprimer the tensor of the strains (virtual or real):

$$\left\{\boldsymbol{\varepsilon}\right\} = \left[B\right].\left\{\boldsymbol{u}\right\} = \left\langle \boldsymbol{u} \right\rangle.\left[B\right]^T \text{ and } \left\{\tilde{\boldsymbol{\varepsilon}}\right\} = \left[B\right].\left\{\tilde{\boldsymbol{u}}\right\} = \left\langle \tilde{\boldsymbol{u}} \right\rangle.\left[B\right]^T \quad (55)$$

One obtains then for the matrix relating to the bilinear form:

$$[a] = \left\langle \tilde{\boldsymbol{u}} \right\rangle.\int_{\Omega^e} \left[B\right]^T.[\Lambda].[B].d\Omega^e.\left\{\boldsymbol{u}\right\} \quad (56)$$

the matrixes $[B]$ and $[\Lambda]$ contain the possible non-linearity of the behavior and will depend on displacements:

1. $[B]$ is a function of displacements if one is in the situation of the large deformations or the great transformations (large rotations and/or large displacements).
2. $[\Lambda]$ is the matrix of behavior. It becomes dependant on displacements in the case of (and other variables) the nonlinear and/or inelastic behaviors.

In these two cases, the process of resolution of the equations will imply a specific processing (generally, a linearization of the Newton-Raphson type). In a similar way, one will easily obtain the elementary form for the second member.

5.3 Computation of the elementary terms

the elementary terms with calculating are form:

$$\int_{\Omega_e} f\left(\mathbf{u}(\mathbf{x}), \frac{\partial \mathbf{u}(\mathbf{x})}{\partial \mathbf{x}}\right).d\,\boldsymbol{x} \tag{57}$$

Three types of operations are to be carried out:
1.the transformation of derivatives compared to $\mathbf{x}$ in derived compared to $\boldsymbol{\xi}$;
2.the transition of an integration on the real element with an integration on the element of reference,
3.the numerical realization of this integration which is generally made by a formula of squaring.

5.3.1 Transformation of derivatives

the transformation of derivatives is carried out thanks to the jacobian matrix $\boldsymbol{J}$, according to the derivative rule in character string:

$$\frac{\partial u_\alpha}{\partial x_\beta} = \frac{\partial \xi_\gamma}{\partial x_\beta}.\frac{\partial u_\alpha}{\partial \xi_\gamma} = \boldsymbol{J}^{-1}.\left[\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\xi}}\right]^T.\boldsymbol{u}_\alpha^{\mathrm{nod}} \tag{58}$$

where $\boldsymbol{u}_\alpha^{\mathrm{nod}}$ is the vector of the nodal values of the component α of displacement.

5.3.2 Change of field of integration

the transition to integration on the element of reference is carried out by multiplying the intégrande by the determinant of the jacobian matrix, called jacobian:

$$\int_{\Omega_e} f\left(\mathbf{u}(\mathbf{x}), \frac{\partial \mathbf{u}(\mathbf{x})}{\partial \mathbf{x}}\right).d\,\boldsymbol{x} = \int_{\Omega_r} f\left(\boldsymbol{u}(\boldsymbol{\xi}), \frac{\partial \boldsymbol{u}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\right).\det(\boldsymbol{J}(\boldsymbol{\xi})).d\,\boldsymbol{\xi} \tag{59}$$

the transition of the element of reference to the real element implies the bijectivity of the transformation τ. It is thus necessary $\det(\boldsymbol{J}(\boldsymbol{\xi}))\neq 0$, which implies that the element should not be turned over or degenerate (for example it is not necessary that the quadrangle degenerates into triangle).

5.3.3 Numerical integration

In certain typical cases, one can calculate the integrals analytically. For example, for a triangle in two dimensions, the Jacobian are constant on the triangle, and the intégrandes are brought back to students' rag processions which one can integrate exactly thanks to the formula of numerical integration known as “of Gauss”[2]:

$$\int_0^1 \int_0^{1-\xi} \xi_1^\alpha.\xi_2^\beta.d\,\xi_1.d\,\xi_2 = \frac{\alpha!\,\beta!}{(\alpha+\beta+2)!} \tag{60}$$

However, these typical cases are rare, and one prefers to evaluate the integrals numerically by calling on formulas of squaring. Those give an approximation of the integral in the form of a balanced sum of the values of the intégrande in a certain number of points of the element called points of integration:

$$\int_{\Omega_r} g(\boldsymbol{\xi}).d\,\boldsymbol{\xi} \approx \sum_{g=1}^{r} \omega_g.g(\xi_g) \tag{61}$$

2 abuse language, one frequently calls the numerical diagrams of integration “diagrams of Gauss” although there are several kinds (Hammer for the triangle, Gauss-Radau, Newton-Dimensions, etc).

the scalars ω_g are called the weights of integration, and the coordinated ξ_g are the coordinates of r the points of integration in the element of reference.
In the integration methods of Gauss, the points and weights of integration are given so as to integrate exactly polynomials of a nature given. It is this kind of method which one uses in *Code_Aster*, the points of integration are called then Gauss points.
The number of Gauss points selected makes it possible to integrate exactly in the element of reference. In fact, because of the possible non-linearity of the geometrical transformation or the spatial dependence of the coefficients (for example for elements deformed or of second order), integration is not exact in the real element.
For each element Ω_e, one knew to calculate the terms known as elementary: elementary matrix $\mathbf{A}^e$ and elementary vector $\mathbf{L}^e$. The matrix A and the vector L are obtained by a procedure that one calls the assembly of the elementary terms.
If one regains the elementary shape of stiffness:

$$[a]=\int_{\Omega^e}\{\sigma\}.\langle\varepsilon\rangle.d\Omega^e \tag{62}$$

numerical integration implies that one evaluates the stresses and the strains at the points of integration:

$$[a]=\int_{\Omega^e}\{\sigma\}.\langle\varepsilon\rangle.d\Omega^e\approx\sum_{g=1}^{r}\omega_g.\{\sigma(\xi_g)\}.\langle\varepsilon(\xi_g)\rangle \tag{63}$$

What means that the stresses and the strains are most exact (or the least false) at the points of integration (fields known as “ELGA” in *Code_Aster*). The simple fact of extrapolating these values with the nodes for the display introduces an error. It is besides about a method evaluation of the error, called error indicator of Zhu-Zienkiewicz.
In elasticity 2D, a triangle exhibant a jacobian constant, only one Gauss point is sufficient to integrate exactly the terms of the matrix and the second member (if it is constant).
The cost computation increases with the number of points of integration, particularly for the nonlinear constitutive laws. For example, a hexahedron with 27 nodes needs 27 Gauss points to integrate the quantities. It thus arrives frequently that one “under-just”, i.e. that one uses less points of integration than the required minimum, thus making a mistake that one will possibly compensate by a finer mesh. Besides this systematic error, this under-integration must be made with precaution because it can produce defaults of row of the matrix and thus make the system linear noninvertible.

6 Resolution of the matric system

One thus obtains a linear system to solve:

$$\langle\tilde{\boldsymbol{u}}\rangle.[\boldsymbol{A}].\{\boldsymbol{u}\}+\langle\tilde{\boldsymbol{u}}\rangle.\{\boldsymbol{L}\}=0 \tag{64}$$

Whatever the field of virtual displacements, therefore:

$$[\boldsymbol{A}].\{\boldsymbol{u}\}=\{\boldsymbol{L}\} \tag{65}$$

6.1 Imposition of the kinematical boundary conditions

the processing of the kinematical boundary conditions of the type $\mathbf{u}=\mathbf{u}^{\mathbf{D}}$ is done in two different ways:

1. The “kinematical” method (AFFE_CHAR_CINE in Code_Aster) consists in modifying the matrix and the second member. This method is fast and does not introduce additional variables. On the other hand, it is not general and does not allow to apply complex limiting conditions of the style $\sum \boldsymbol{u}_i.a_i=\boldsymbol{u}^{\boldsymbol{D}}$.
2. The method by dualisation (AFFE_CHAR_MECA in Code_Aster) consists in introducing a vector of multipliers (or parameters) of Lagrange $\boldsymbol{\lambda}$, which increases the number of unknowns but makes it possible to treat all the cases.

$$\begin{cases}[\boldsymbol{A}].\{\boldsymbol{u}\}+[\boldsymbol{Q}]^T\{\boldsymbol{\lambda}\}=\{\boldsymbol{L}\}\\ [\boldsymbol{Q}].\{\boldsymbol{u}\}=\{\boldsymbol{u}^{\boldsymbol{D}}\}\end{cases} \tag{66}$$

6.2 Resolution

the linear system can be solved by a certain number of numerical methods. The methods used in *Code_Aster* are factorization LDL^T per blocks, the mulitfrontal method (or its equivalent with swivelling, MUMPS), and the preconditioned conjugate gradient.
The methods of resolution are divided into three categories:
- The direct methods which solve exactly (with the numerical errors near)
- the iterative methods
- the hybrid methods, very much used in the methods of decomposition of fields. (see method FETI [R6.01.03]).

The matrixes resulting from the finite element method are very hollow (they comprise a majority of null terms). In practice, on systems of standard size (a few tens of thousands of equations), the density of non-zero terms seldom exceeds the 0.01%. They are thus stored in form digs (or “sparse”) and take little core in memory. A contrario, the matrixes are not built to be used effectively with the mathematical libraries of programs optimized dedicated to the full matrixes (booksellers BLAS for example). Solvers are thus developed specifically for these problems.
A direct solver has as a principle breaking up the matrix into a product of particular matrixes of form. For example, decomposition LDL^T :

$$[A]=[L].[D].[L]^T \quad (67)$$

Where the matrix D is diagonal and the matrix L is triangular lower. This decomposition is valid only for the symmetric matrixes. If it is not the case, other decompositions should be used.
The principle is the following:
•From the initial matrix (very hollow), one builds a product of remarkable matrixes. It is the operation of *factorization*.
•These remarkable matrixes make it possible to solve the very fast problem of way. It is the phase of *descent-increase.*

The phase of factorization is most expensive. For the most spread decompositions, the cost machine is in n^3 where n is the number of equations. The cost report will depend on the profile of the matrix (of classification of the finite elements). Automatic processes seek to optimize this classification to have a structure as compact as possible. Even with this optimization, it is frequent that the factorized matrix take several hundreds of times, even several thousands of times more memory than the initial matrix. The direct solvers thus consume much memory and that becomes crippling about it from several hundreds of thousands of degrees of freedom, even on the most powerful machines. On the other hand, these direct methods are particularly robust. The problems in structural mechanics and of solids very often lead to matrixes with a bad conditioning (it is particularly the case of all the last numerical innovations which use mixed methods with many Lagrange multipliers).
When it is possible, iterative methods whose principle consists in finding an approximation of the reverse of the matrix and to proceed then to an iterative resolution, not by step, which uses only products matrix-vectors, very effective and inexpensive in memory are preferentially used.
However, these iterative methods have several defaults:
•They are less robust than direct methods, particularly when conditioning is bad
•the methods of preconditionning are very numerous and there are some as much as different problems (even several possible by problem). What obliges the user to juggle with the various methods, without never being assured to obtain result at the end.
•They are iterative methods, which implies stopping criteria of the process, and thus a parameter to be managed but also problems of office plurality of round-offs.

Finally the hybrid methods try to reconcile the advantages of the two approaches. Generally, one uses them in the methods of decomposition of fields, where each field is treated by a direct solver while the problem of interface is solved by an iterative solver. One can quote the method FETI (and its alternatives, to see [R6.01.03]) or the variations of the method LATIN. It is about an extremely active field of search.

7 Organization of a computation by finite elements in *the Code_Aster*

One very briefly describes how and at which place the aspects evoked in this document are established in *Code_Aster*.

7.1 Notion of finite element in *Code_Aster*

A kind of finite element is defined by:
- a kind of mesh
- nodes list
- of the shape functions
- of the computation options

an element in the mesh is defined by a kind of mesh, a geometry (coordinated nodes) and a topology (ordered list of the nodes). It is the type of modelization chosen in the command file which makes it possible to assign to each mesh of the mesh a kind of finite element. The command AFFE_MODELE [U4.22.01] assigns to each mesh a kind of finite element corresponding to the modelization specified for this mesh.

Notice important:

One should not forget to assign of the finite elements to meshes edge which one needs to impose the boundary conditions and loadings, and that one will have taken care to create during the fabrication of the mesh.

Operator AFFE_CHAR_MECA [U4.44.01], which affects boundary conditions and loadings, also will create of the finite elements, for example the finite elements which will carry the degrees of freedom of LAGRANGE used in the dualisation of the boundary conditions [R3.03.01].
Operator AFFE_CARA_ELEM [U4.42.01] allows to define additional characteristics for some element types: for example, the thickness of the shells, directional sense of the beams, mass matrixes and of stiffness of the discrete elements.
A computation option indicates the elementary type of computation that the element is able to calculate. For example RIGI_MECA relates to the computation of the elementary matrix of mechanical stiffness:

$$A^e_{\alpha\beta} = \int_{\Omega^e} \Lambda_{ijkl} . \varepsilon_{ij}(N^e_\alpha(\mathbf{x})) . \varepsilon_{kl}(N^e_\beta(\mathbf{x})) . d\Omega^e \quad (68)$$

the "data" of this option are the geometry Ω^e and the material Λ, supplemented by the temperature if the material depends on it.
Let us recall that to apply the loadings of border, one uses edge individuals of the finite elements, and not the borders of the finite elements of volume (3D) or surface (2D).

Note:

A developer can sometimes have the choice between creating a new finite element or adding a computation option to an existing element; the choice between these two solutions in general takes account of criteria of data-processing facility (e.g. elements under - integrated).

7.2 Initializations of the elements

the use of elements of reference makes it possible once and for all to carry out a certain number of computations at the beginning of the execution. One defines, for each type of element of reference:
- The number of nodes and their coordinates;
- The number of families of Gauss points;
- The number of Gauss points;
- Weights of integration ω_g ;
- Values of the shape functions to Gauss points $N_i(\boldsymbol{\xi}_\mathbf{g})$;
- Values of derivatives of the shape functions to Gauss points $\frac{\partial N_i(\boldsymbol{\xi}_\mathbf{g})}{\partial \boldsymbol{\xi}}$.

For a given element, one inevitably does not integrate all the elementary terms with the same number of Gauss points: for example, one uses in general of Gauss points for the mass matrix than for the stiffness matrix, because the products of shape functions are of degree higher than the products of their derivatives. Another example is the under-integration used in certain cases. One calls of Gauss points family each whole of Gauss points likely to be used.

7.3 Computation of the elementary terms

During the computation of the elementary terms (in the routines YOU….), one carries out for each Gauss point the following operations:
- Computation of derivatives of the shape functions on the real element starting from the coordinates of the nodes of the element and derivatives of the shape functions on the element of reference;
- Computation of the jacobian matrix;
- Recovery of the weight of integration multiplied by the Jacobian at the Gauss point considered;
- Evaluating of the intégrande (according to the calculated option).

The elementary term is calculated by sum on Gauss points while balancing by the weights of integration.

7.4 Total resolution

the total resolution takes place in routines OP…. high level corresponding to the commands user (MECA_STATIQUE [U4.51.01], STAT_NON_LINE [U4.51.03], THER_LINEAIRE [U4.54.01], etc).

8 Bibliography

1) P.G. Ciarlet, “The finite element method for elliptic problems”, Studies in Applied Mathematics, North Holland, 1978.

2) A. Ern, J. - L. Guermond, “Finite elements: theory, applications, put in work”, Springer, 2002.

3) G. Dhatt, G. Touzot, E. Lefrançois, “Finite element method: a presentation”, Hermes, 2004.

9 Description of the versions of the document

Version Aster	Author (S) Organization (S)	Description of the modifications
3	I.VAUTIER	initial Text
10.2	M.Abbas	partial Rewriting, “anonymization” of the concepts compared to the mechanical