La méthode des éléments finis isoparamétriques

Résumé :

Ce document présente les fondements des éléments finis iso-paramétriques introduits dans *Code_Aster* pour la modélisation des milieux continus 2D et 3D. On rappelle tout d'abord le passage d'une formulation forte à une formulation variationnelle, puis on détaille la discrétisation par éléments finis : utilisation d'un élément de référence, calcul des fonctions de forme et évaluation des termes élémentaires. On décrit également brièvement le principe de l'assemblage de ces termes et de l'imposition des conditions aux limites, et on évoque les méthodes de résolution matricielle utilisées. Enfin sont exposées les principales étapes d'un calcul par éléments finis tel qu'il est conçu et implanté dans *Code_Aster*.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Table des matières

1 Introduction....3
2 Obtention d'une formulation variationnelle....3
2.1 Modélisation du problème physique – Principes et notation....3
2.2 Équations du système....3
2.3 Méthode des résidus pondérés – Formulation intégrale forte....4
2.4 Formulation intégrale faible....4
3 Méthode des éléments finis....5
3.1 Principes généraux....5
3.2 Approximation de la géométrie....5
3.2.1 Principe....5
3.2.2 Éléments de référence....6
3.2.3 Fonctions d'interpolation géométrique....6
3.2.4 Matrice jacobienne de la transformation....7
3.3 Représentation des inconnues....7
3.3.1 Approximation nodale....7
3.3.2 Base polynomiale....8
3.3.3 Fonctions de forme....9
3.3.4 Correspondance entre base polynomiale et fonctions de forme....10
3.4 Résultats d'existence et d'unicité....10
4 Méthode de Ritz....10
5 Construction du système matriciel....11
5.1 Nouvelle notation (notation de Voigt)....11
5.2 Système discrétisé....12
5.3 Calcul des termes élémentaires....14
5.3.1 Transformation des dérivées....14
5.3.2 Changement de domaine d'intégration....14
5.3.3 Intégration numérique....14
6 Résolution du système matriciel....15
6.1 Imposition des conditions aux limites cinématiques....15
6.2 Résolution....16
7 Organisation d'un calcul par éléments finis dans le Code_Aster....16
7.1 Notion d'élément fini dans Code_Aster....17
7.2 Initialisations des éléments....17
7.3 Calcul des termes élémentaires....18
7.4 Résolution globale....18
8 Bibliographie....18
9 Description des versions du document....18

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

1 Introduction

La méthode des éléments finis est employée dans de nombreux domaines scientifiques pour résoudre des équations aux dérivées partielles. Elle permet de construire une approximation simple des inconnues pour transformer ces équations continues en un système d'équations de dimension finie, que l'on peut écrire schématiquement sous la forme suivante :

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

où $\{U\}$ est le vecteur des inconnues, $[A]$ une matrice et $\{L\}$ un vecteur.

Dans un premier temps, on transforme les équations aux dérivées partielles en une formulation intégrale (ou formulation ***forte*** du problème), souvent cette première forme intégrale est modifiée (affaiblie) en utilisant la formule de Green (on obtient alors une formulation ***faible***). La solution approchée est cherchée comme combinaison linéaire de fonctions données. Ces fonctions doivent être simples mais assez générales pour pouvoir « bien » approcher la solution. Elles doivent notamment permettre de générer un espace de dimension finie qui soit aussi proche que l'on veut de l'espace de fonctions dans lequel se trouve la solution. A partir de cette idée ancienne (méthode des résidus pondérés), les diverses façons de choisir ces fonctions donnent lieu à différentes méthodes numériques (collocation, méthodes spectrales, éléments finis, etc.).

L'originalité de la méthode des éléments finis est de prendre comme fonctions d'approximation des polynômes qui sont nuls sur presque tout le domaine, et participent donc au calcul seulement au voisinage d'un point particulier. Ainsi, la matrice $[A]$ est très creuse, ne contenant que les termes d'interaction entre « points voisins », ce qui réduit le temps de calcul et la place mémoire nécessaire au stockage. De plus, la matrice $[A]$ et le vecteur $\{L\}$ peuvent être construits par assemblage de matrices et vecteurs élémentaires, calculés localement.

2 Obtention d'une formulation variationnelle

On peut obtenir la formulation variationnelle d'un problème à partir des équations aux dérivées partielles, en multipliant celles-ci par des fonctions tests et en intégrant par parties. En mécanique des solides, la formulation faible alors obtenue est identique à celle donnée par le Principe des Travaux Virtuels et dans le cas conservatif, la minimisation de l'énergie potentielle totale de la structure. Notons cependant que pour certains problèmes, les équations du modèle sont plus faciles à établir dans le cadre variationnel (cas des plaques et des coques par exemple).

2.1 Modélisation du problème physique – Principes et notation

Un système physique est le plus souvent modélisé par des équations aux dérivées partielles qui agissent sur des inconnues $\mathbf{u}$ qui peuvent être :

- Un scalaire comme la température dans les problèmes de thermique;
- Un vecteur comme les déplacements dans les problèmes de mécanique;
- Un tenseur comme les contraintes dans les problèmes de mécanique;

On peut également utiliser plusieurs champs d'inconnues simultanément, reliés par des équations aux dérivées partielles. Ce sont des problèmes *couplés*. Dans Code_Aster, on peut citer comme exemple les problèmes de thermo-hydro-mécanique qui couplent déplacements, pression et température.
Les champs d'inconnues sont paramétrés par :

- L'espace, qui peut être décrit par un système de coordonnées cartésiennes ou n'importe quel autre type de paramétrisation. Dans la suite du document, on le notera x ;
- Le temps, noté t ;

2.2 Équations du système

Un système physique continu peut être représenté par un *système* d'équations aux dérivées partielles que l'on écrira dans le domaine Ω :

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

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

Ce système est associé aux conditions aux limites sur la frontière Γ du domaine Ω :

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

L'opérateur différentiel peut s'exprimer sur plusieurs équations aux dérivées partielles. On pourrait écrire :

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

$L_i(\mathbf{u})$ est un opérateur différentiel agissant sur le vecteur des inconnues $\mathbf{u}$. De manière plus générale, l'opérateur différentiel $L_i(\mathbf{u})$ s'écrit en fonction des inconnues et de leurs dérivées partielles :

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

Un tel opérateur est dit d'ordre m en espace et d'ordre p en temps. S'il ne dépend pas du temps (et de ses dérivées), on dit que le problème est stationnaire. Dans la suite du document on ne considérera que les problèmes stationnaires.

2.3 Méthode des résidus pondérés – Formulation intégrale forte

On va définir le résidu $\mathbf{R}(\mathbf{u})$ comme étant la quantité s'annulant quand $\mathbf{u}$ est la solution du problème physique :

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

La méthode des résidus pondérés consiste :
1/ A construire une solution approchée $\mathbf{u}$ par la combinaison linéaire de fonctions judicieusement choisies

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

Où $\boldsymbol{\phi}_i(\boldsymbol{x})$ sont les fonctions de forme de l'approximation et c_i des coefficients à identifier.
2/ A résoudre le système sous forme intégrale :

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

Nous avons utilisé les mêmes fonctions de pondération pour le système principal et pour les conditions limites, mais ce n'est pas obligatoire. $\mathbf{P}(\mathbf{u})$ sont les fonctions de pondération appartenant à un ensemble de fonctions E_P. La solution $\mathbf{u}$ appartient à l'espace E_u des fonctions « suffisamment » régulières (dérivables jusqu'à l'ordre *m*).
Le choix des fonctions de pondération $\mathbf{P}(\mathbf{u})$ permet de dégager plusieurs méthodes :

- Si la fonction $\mathbf{P}(\mathbf{u})$ est une distribution de Dirac, on obtient la méthode de collocation par points.
- Si la fonction $\mathbf{P}(\mathbf{u})$ est constante sur des sous-domaines, on obtient la méthode de collocation par sous-domaines.
- Si les fonctions de pondération $\mathbf{P}(\mathbf{u})$ utilisent les mêmes fonctions de forme $\boldsymbol{\phi}_i(\boldsymbol{x})$ que l'approximation de la solution (7), on obtient la méthode de Galerkin.

On obtient ainsi la ***forme intégrale forte***.

2.4 Formulation intégrale faible

La formulation intégrale (8) requiert des espaces de fonction dérivables à l'ordre m pour E_u. La formulation faible consiste à réaliser une intégration par parties (par application de la formule de Green) du système (8). En contrepartie on augmente les exigences de régularité sur les fonctions de pondérations $\mathbf{P}(\mathbf{u})$. La formule de Green s'énonce comme suit :

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

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

où $\boldsymbol{n}$ est la normale sortante à la frontière Γ du domaine.

3 Méthode des éléments finis

3.1 Principes généraux

La recherche d'une fonction approchée idoine sur tout le domaine devient difficile dans le cas général d'une géométrie de forme quelconque. L'idée de la méthode des éléments finis est donc de construire cette approximation en deux temps :

- Identifier des sous-domaines Ω_e géométriquement simples qui pavent le domaine ;
- Définir une fonction approchée sur chaque sous-domaine ;

On pressent donc un certain nombre de caractéristiques de cette construction :

- Le pavage du domaine Ω_e par les sous-domaines Ω_e doit être aussi précis que possible ;
- La fonction approchée sur le sous-domaine doit respecter des conditions de continuité entre les différents sous-domaines ;
- La fonction approchée sur le sous-domaine doit avoir des propriétés cohérentes avec les conditions de dérivabilité et en rapport avec la description physique de la solution (ce qui peut impliquer d'utiliser une formulation affaiblie par exemple).

3.2 Approximation de la géométrie

3.2.1 Principe

On identifie les N_e sous-domaines (ou *éléments*) Ω_e qui pavent l'espace Ω du solide :

$$\Omega = \sum_{e=1}^{N_e} \Omega_e \quad (10)$$

Notons $x_{\alpha=1,3}$ les coordonnées d'un point $\boldsymbol{x}$ dans le repère absolu. La géométrie du sous-domaine est construite avec une approximation nodale, soit pour un *élément* avec N_{nd} nœuds :

$$\boldsymbol{x}^e = \sum_{i=1}^{N_{nd}} \boldsymbol{x}_i^e . \bar{N}_i^e \text{ ou } x_\alpha^e = \sum_{i=1}^{N_{nd}} x_{\alpha,i}^e . \bar{N}_i^e \quad (11)$$

Ce pavage (maillage) est une opération pouvant être complexe, surtout en 3D. Il existe des algorithmes généraux pour mailler. On utilise des triangles ou des quadrangles en 2D et des tétraèdres ou des hexaèdres en 3D (plus quelques éléments servant de raccords). Les triangles et tétraèdres donnent ce qu'on appelle des maillages *libres* , les quadrangles et les hexaèdres forment des maillages *réglés* . Les maillages libres sont relativement faciles à construire grâce à des techniques largement éprouvées : cellules de Voronoï construisant une triangulation de Delaunay ou méthodes de propagation (méthodes dites frontales), les maillages réglés sont beaucoup plus délicats à générer. Le maillage induit nécessairement une erreur de discrétisation géométrique Par exemple, sur la figure (1), on voit qu'une frontière courbe n'est qu'imparfaitement approchée par des éléments linéaires.

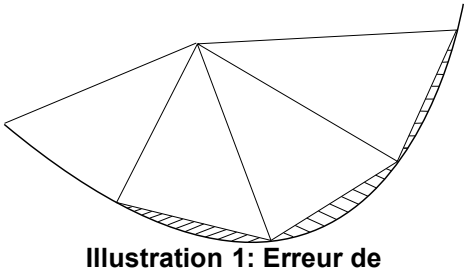


Illustration 1: Erreur de discrétisation géométrique

De même le maillage doit être conforme : pas de trous ou de recouvrement (voir figure (2)).

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

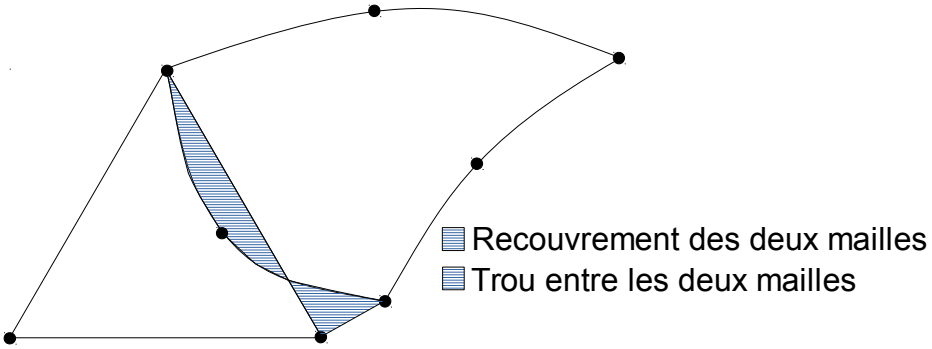


Illustration 2: Non-conformité du maillage

Pour respecter cette condition de *conformité*, il suffit de deux règles :

1. Chaque élément doit être défini de manière unique à partir des coordonnées de ses nœuds géométriques (et pas ceux de ses voisins !);
2. La frontière d'un élément doit être définie de manière unique à partir des nœuds de ces frontières, ces nœuds étant communs entre les éléments partageant cette frontière.

Ces conditions de conformité sont une différence importante par rapport aux volumes finis qui n'ont pas ces exigences. Le pavage du domaine permet d'appliquer la méthode des éléments finis sur des géométries complexes, au contraire des méthodes par différences finies. Le pavage géométrique du domaine induit une première erreur : il n'est pas possible, dans le cas général, de représenter une géométrie réelle par un maillage par des polygones réguliers, en particulier sur la frontière du domaine.

« ***Un beau maillage est un bon maillage*** »

3.2.2 Éléments de référence

Le calcul des fonctions de forme pour un élément quelconque peut être assez compliqué. C'est pourquoi on préfère souvent se ramener à un élément dit de référence, à partir duquel on peut générer tous les éléments d'une même famille par une transformation géométrique. Les fonctions de forme sont alors calculées sur cet élément générique noté Ω_r, et le transport des grandeurs sur l'élément réel Ω_e est effectué grâce à la connaissance de la transformation géométrique.

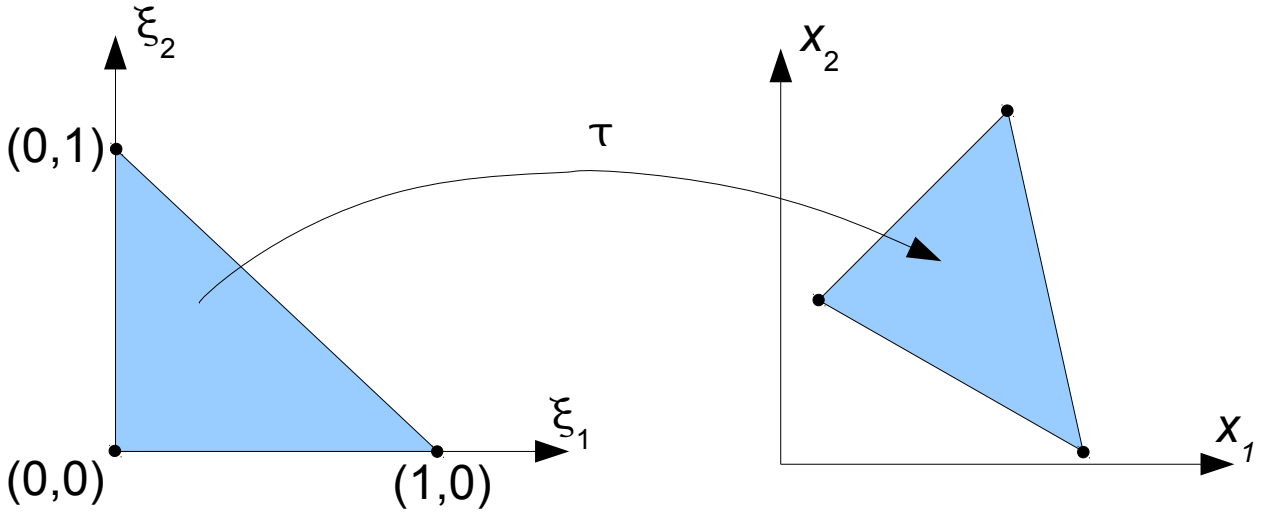


Illustration 3: Passage de l'espace de référence à l'espace réel

Les points de l'élément de référence seront décrits en termes de coordonnées paramétriques $\xi_{\alpha=1,3}$. La transformation τ doit être bijective et transformer les sommets et côtés de l'élément de référence en sommets et côtés de l'élément réel :

$$\xi_\alpha \overset{\tau}{\rightarrow} x_\alpha \tag{12}$$

3.2.3 Fonctions d'interpolation géométrique

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

La géométrie de l'élément est donc approchée par le biais de fonctions d'interpolation géométrique. Ces fonctions notées $\bar{N}(\boldsymbol{\xi})$ sont définies sur l'élément de référence ; elles permettent de connaître les coordonnées x_α d'un point quelconque de l'élément réel à partir de ses coordonnées ξ_α de son antécédent dans l'élément de référence et des coordonnées x_α^i des nœuds (de numéro local I) de l'élément réel :

$$\boldsymbol{x}^e = \sum_{i=1}^{N_{nd}} \boldsymbol{x}_i^e . \bar{\boldsymbol{N}}_i^e \text{ ou } x_\alpha^e = \sum_{i=1}^{N_{nd}} x_{\alpha,i}^e . \bar{N}_i^e \tag{13}$$

3.2.4 Matrice jacobienne de la transformation

La jacobienne de la transformation est la matrice des dérivées partielles des coordonnées réelles x_α par rapport aux coordonnées ξ_α dans l'élément de référence :

$$J_{\alpha\beta} = \frac{\partial x_\alpha}{\partial \xi_\beta} \tag{14}$$

En tenant compte de la définition des coordonnées x_α en fonction des coordonnées $x_{\alpha,i}$ des nœuds, on obtient une expression équivalente de la matrice jacobienne :

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

Où $\frac{\partial \bar{N}_i}{\partial \xi_\beta}$ sont les termes de la matrice $\left[\frac{\partial \bar{\boldsymbol{N}}}{\partial \boldsymbol{\xi}}\right]^T$, dont le nombre de lignes est le nombre de directions de l'espace, et le nombre de colonnes le nombre de nœuds de l'élément. Notons que la matrice $\left[\frac{\partial \bar{\boldsymbol{N}}}{\partial \boldsymbol{\xi}}\right]^T$ ne dépend que de la définition de l'élément de référence et non de celle de l'élément réel. Le déterminant de la matrice jacobienne, utile dans les calculs qui vont suivre, s'appelle le jacobien de la transformation géométrique. Il est non nul lorsque la transformation τ qui fait passer de l'élément de référence à l'élément réel est bijective, et positif lorsque τ respecte l'orientation de l'espace.

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

3.3 Représentation des inconnues

Pour résoudre le problème, on considère une approximation par éléments finis d'un champ inconnu. Les espaces E_P et E_u sont représentés par des espaces discret E^h. Il y a deux façons équivalentes de représenter les inconnues dans un élément : par les coefficients de leur approximation polynomiale, ou par leurs valeurs nodales. Ces deux possibilités correspondent aux deux manières complémentaires de définir un élément : par la donnée d'une base de monômes, ou par la donnée des fonctions de forme associées aux nœuds. De manière générale, on construit la fonction approchée en écrivant la relation linéaire suivante sur chaque élément :

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

Où les $\boldsymbol{\phi}_i^e(\boldsymbol{\xi})$ sont des fonctions linéaires indépendantes. Ils constituent la ***base*** de l'approximation, les paramètres généraux de l'approximation étant les coefficients a_i.

3.3.1 Approximation nodale

La première idée de la méthode des éléments finis est de construire une approximation de type nodale pour laquelle les coefficients $u_i = a_i$ correspondent à la solution en ces nœuds :

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

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

On obtient alors une approximation nodale avec $\boldsymbol{N}_i^e(\boldsymbol{\xi})$ les fonctions d'interpolation sur l'élément de référence. Sur chacun de ces sous-domaines on construit une fonction approchée différente d'un sous-domaine à l'autre. L'approximation éléments finis est *élémentaire* car la fonction ne dépend que des valeurs nodales constituant l'élément :

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

Un élément est *isoparamétrique* lorsqu'il est basé sur des interpolations identiques pour sa géométrie et ses inconnues : $\bar{N}(\boldsymbol{\xi}) = N(\boldsymbol{\xi})$.
Pour assurer la continuité de la solution sur l'élément et, éventuellement, la continuité de ses dérivées, il faut que les fonctions $\boldsymbol{N}_i^e(\boldsymbol{\xi})$ soient continues et, éventuellement, à dérivées continues.
De même si l'on veut assurer la continuité de la solution et de ses dérivées aux frontières des éléments (conformité de l'approximation), il faut que la solution et ses dérivées dépendent de manière unique des variables nodales des nœuds de la frontière.

3.3.2 Base polynomiale

La façon la plus simple de définir un élément est de choisir une base polynomiale composée d'un certain nombre de monômes indépendants. Pour une inconnue donnée, le nombre de monômes utilisés doit être égal au nombre de variables nodales, c'est-à-dire au nombre de nœuds utilisés pour représenter l'inconnue. On définit généralement la base polynomiale sur l'élément de référence ; elle contient des monômes de la forme $\xi_1^{\gamma} . \xi_2^{\delta} . \xi_3^{\varepsilon}$, où γ , δ et ε sont des exposants entiers positifs ou nuls. Le degré d'un tel monôme est l'entier $\gamma+\delta+\varepsilon$. La base est dite complète de degré n lorsque tous les monômes de degré n sont présents. Dans certains cas, on emploie des bases incomplètes. On note $P_p(\boldsymbol{\xi})$ le $p^{ième}$ monôme de la base (qui en comprend m). Les composantes du vecteur déplacement $\mathbf{u}(\boldsymbol{\xi})$ dans l'élément sont alors données par la formule :

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

On notera Π la matrice donnant les valeurs prises par les monômes de la base polynomiale sur les nœuds de l'élément de référence :

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

où p est le numéro d'ordre du monôme dans la base, I le numéro du nœud localement à l'élément et $\boldsymbol{\xi}_I$ les coordonnées du nœud I dans l'élément de référence. Cette matrice est carrée, sa dimension est le carré du nombre de nœuds de l'élément.
Au nœud I le déplacement u_α^I vaut :

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

On distingue trois grands types d'éléments finis fréquemment utilisés :

- les éléments finis de Lagrange qui reposent sur des bases polynomiales complètes et différents types de géométries (symplectiques pour les triangles et les tétraèdres, à structure tensorielle pour les quadrangles et les hexaèdres ou de type prismatique);
- les éléments finis de type Serendip, qui sont des éléments finis de Lagrange avec des bases incomplètes;
- les éléments finis d'Hermite, de haute précision, qui utilisent les inconnues nodales et leurs dérivées;

Éléments finis de Lagrange symplectiques

Pour déterminer si une base polynomiale est complète avec les éléments symplectiques, il suffit d'utiliser le triangle de Pascal :

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Ordre					
Constant			1		
Linéaire		ξ_1		ξ_2	
Quadratique	$(\xi_1)^2$		$\xi_1.\xi_2$		$(\xi_2)^2$
Cubique	$(\xi_1)^3$	$(\xi_1)^2.\xi_2$		$\xi_1.(\xi_2)^2$	$(\xi_2)^3$
Ordre 4	$(\xi_1)^4$	$(\xi_1)^3.\xi_2$	$(\xi_1)^2.(\xi_2)^2$	$\xi_1.(\xi_2)^3$	$(\xi_2)^4$

Une base polynomiale complète d'ordre deux comporte six monômes : $\{1;\xi_1;\xi_2;\xi_1^2;\xi_2^2;\xi_1.\xi_2\}$ et donc l'élément géométrique support sera un triangle à six nœuds.

Éléments finis de Lagrange à structure tensorielle

Pour décrire des éléments finis quadrangulaires (ou héxaèdriques), il suffit de prendre des polynômes complets de l'ordre donné et en faire le produit.

Ordre	Constant	Linéaire	Quadratique	Cubique
Constant	1	ξ_1	$(\xi_1)^2$	$(\xi_1)^3$
Linéaire	ξ_2	$\xi_1.\xi_2$	$(\xi_1)^2.\xi_2$	$(\xi_1)^3.\xi_2$
Quadratique	$(\xi_2)^2$	$\xi_1.(\xi_2)^2$	$(\xi_1)^2.(\xi_2)^2$	$(\xi_1)^3.(\xi_2)^2$
Cubique	$(\xi_2)^3$	$\xi_1.(\xi_2)^3$	$(\xi_1)^2.(\xi_2)^3$	$(\xi_1)^3.(\xi_2)^3$

Une base polynomiale « sur »-complète d'ordre deux pour un élément quadrangulaire comporte neuf monômes : $\{1;\xi_1;\xi_1^2;\xi_2;\xi_2^2;\xi_1.\xi_2;\xi_1.\xi_2^2;\xi_1^2.\xi_2;\xi_1^2.\xi_2^2\}$, ce qui signifie neuf nœuds. Un tel élément comporte des termes d'ordre 3 et 4.

Éléments finis de Serendip

Les éléments de Serendip, pour un polynôme d'ordre s, excluent les termes croisés de degré supérieur à $s+1$ pour ne pas avoir de nœuds à l'intérieur des éléments. Par exemple, pour un élément de Serendip d'ordre deux, les monômes seront $\{1;\xi_1;\xi_1^2;\xi_2;\xi_2^2;\xi_1.\xi_2;\xi_1.\xi_2^2;\xi_1^2.\xi_2\}$, soit huit nœuds.

3.3.3 Fonctions de forme

Une façon équivalente de définir un élément fini est de donner, pour chaque inconnue, l'expression des fonctions de forme de l'élément. Pour une inconnue scalaire donnée (composante du déplacement selon y par exemple), il y en a autant que de nœuds où l'inconnue doit être calculée. Dans beaucoup de cas, on utilise les mêmes fonctions de forme pour toutes les composantes d'un vecteur inconnu, mais ce n'est pas obligatoire. Dans ce qui suit, on supposera cependant pour simplifier les écritures que c'est le cas.

Les fonctions de forme peuvent être définies sur l'élément réel Ω_e : on les note alors $N^e(\mathbf{x})$, elles dépendent de la géométrie de l'élément réel, et sont donc différentes d'un élément à l'autre. Il est plus simple de les exprimer sur l'élément de référence, ce qui donne les fonctions $N(\boldsymbol{\xi})$ indépendantes de la géométrie de l'élément réel. Rappelons que ces fonctions sont polynomiales sur l'élément, et que la fonction de forme associé à un nœud donné y prend la valeur un, alors qu'elle s'annule en tous les autres nœuds de l'élément. Les inconnues s'expriment alors comme combinaison linéaire des fonctions de forme, les coefficients $u_{\alpha,i}$ de la combinaison étant appelés les variables nodales :

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

En utilisant la transformation τ entre l'élément de référence et l'élément réel :

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

On a :

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

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

3.3.4 Correspondance entre base polynomiale et fonctions de forme

On a deux relations. La première provient de l'approximation de la solution par une base polynomiale :

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

La seconde est l'approximation nodale :

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

La matrice donnant les valeurs prises par les monômes de la base polynomiale sur les nœuds de l'élément de référence :

$$\Pi_{Ip}=P_p(\xi_I) \quad (28)$$

En un nœud I, on a écrit l'approximation polynomiale suivante;

$$u_{I,\alpha}=a_{\alpha,p}.\Pi_{Ip} \quad (29)$$

En injectant l'équation (29) dans l'expression nodale (27), on obtient :

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

Par comparaison avec l'approximation polynomiale (26), on en déduit la relation suivante entre la base polynomiale et les fonctions de forme :

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

En pratique, on trouvera dans la littérature les écritures des fonctions de forme nodales pour les éléments les plus courants, selon le choix de la base polynomiale.

3.4 Résultats d'existence et d'unicité

On peut écrire le problème de manière plus abstraite :

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

E_u et E_v sont des espaces vectoriels de fonctions définies sur Ω. Ce sont des espaces de Hilbert.
$a(\mathbf{u},\mathbf{v})$ est une forme bilinéaire sur $E_u\times E_v$ (on a supposé que $\mathbf{L}(\mathbf{u})$ représente un problème physique linéaire par rapport à $\mathbf{u}$).
$f(\mathbf{v})$ est une forme linéaire et continue sur E_v.
Pour établir les conditions d'existence et d'unicité, on applique le théorème de Lax-Milgram. Dans un premier temps, on suppose que la solution appartient au même espace que les fonctions test $E_u=E_v$
Si la forme $a(\mathbf{u},\mathbf{v})$ est coercive c'est-à-dire :

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

Alors le problème :

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

admet une et une seule solution.

4 Méthode de Ritz

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

La méthode de Galerkine, dans certains cas, équivaut à rendre stationnaire une fonctionnelle. C'est le cas si la forme bilinéaire $a(\mathbf{u},\mathbf{v})$ est symétrique et positive :

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

Dans ce cas le problème (34) admet une et une seule solution $\mathbf{u}$ minimise sur E_u la fonctionnelle suivante :

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

Du point de vue mécanique, celui signifie que le principe des puissances virtuelles peut aussi s'écrire comme la minimisation d'une grandeur scalaire : l'énergie totale de la structure. Cette manière d'écrire l'équilibre est très fréquemment employée. Nous allons ici en présenter quelques résultats.

Nous rappelons d'abord qu'une fonctionnelle est une fonction d'un ensemble de fonctions (et de ses dérivées). On écrira π cette fonctionnelle. On se limitera aux formulations en déplacement, sachant qu'il en existe d'autres. Dans ce cas, la fonctionnelle π s'écrira :

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

Pour les problèmes conservatifs, on peut démontrer qu'écrire que la première variation de π est nulle (condition de stationnarité de la fonctionnelle) équivaut à appliquer le principe des travaux virtuels, ou encore à utiliser la méthode de Galerkine en prenant les déplacements virtuels comme fonction de pondération. On appelle que la méthode de Galerkine consiste à partir du problème aux dérivées partielles établissant l'équilibre de la structure, soit :

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

On cherche alors à résoudre le problème sous forme intégrale en utilisant des fonctions de pondération qui sont de la même nature que la solution approchée :

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

Si on choisit comme fonction de pondération la variation des inconnues $\psi=\delta\boldsymbol{u}$ et après avoir intégré par parties une fois, on obtient :

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

Trouver la forme exacte de la fonctionnelle n'est pas immédiat dans le cas général. En mécanique, pour les cas conservatifs, il se trouve que cette fonctionnelle équivaut à l'énergie potentielle totale du système. Après discrétisation de la fonctionnelle (par une approximation éléments finis), on se retrouve avec un système matriciel strictement équivalent à celui de la méthode de Galerkine (ou son principe mécanique équivalent, la méthode des puissances virtuelles).

Intuitivement, on comprend qu'une faible variation $\delta\boldsymbol{u}$ de la solution est un champ qui peut être cinématiquement admissible et qui correspond donc bien aux hypothèses de la méthode des puissances virtuelles.

5 Construction du système matriciel

Nous allons maintenant présenter les différents ingrédients menant à la construction du système matriciel qui permettra de résoudre le problème.

5.1 Nouvelle notation (notation de Voigt)

Afin de bien comprendre la construction des termes discrets dans la méthode des éléments finis, nous allons utiliser une notation plus compacte :

$\langle V \rangle$ est un vecteur ligne

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

$\left\{V\right\}$ est un vecteur colonne
$\left[A\right]$ est une matrice

Ainsi l'interpolation géométrique s'écrit suivant les trois dimensions de l'espace :

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

Ou encore de manière plus compacte sous forme vectorielle :

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

Avec la matrice N des fonctions de forme. En considérant un élément à deux noeuds, on obtient sous forme développée :

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

5.2 Système discrétisé

On se place dans le cas hyperélastique en petites déformations, le problème de mécanique à résoudre on écrit de manière plus compacte :

$$\begin{aligned} &\text{Trouver } \boldsymbol{u} \in E^h \text{ tel que } \forall \tilde{\boldsymbol{u}} \in E^{\hat{h}} \\ &\text{avec } a(\boldsymbol{u}, \tilde{\boldsymbol{u}}) + l(\tilde{\boldsymbol{u}}) = 0 \end{aligned} \quad (44)$$

Avec $a(\boldsymbol{u}, \tilde{\boldsymbol{u}})$ une forme bilinéaire, symétrique qui représente l'énergie potentielle de la structure et $l(\tilde{\boldsymbol{u}})$ le potentiel[1] des efforts volumiques et surfaciques :

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

La discrétisation consiste à choisir une base de l'espace Ω^h et à calculer numériquement les termes de la matrice $\boldsymbol{A}$ et du vecteur $\boldsymbol{L}$. Pour cela, on exprime la forme bilinéaire $a(.,.)$ et la forme linéaire $l(.)$ comme une somme sur des éléments, définis par découpage du domaine de base :

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

Les termes A_{ij}, qui représentent l'interaction entre deux degrés de liberté i et j sont construits en *assemblant* (l'opération notée $\sum_{\text{éléments } \Omega^e} (\ldots)$) les contributions provenant de chacun des éléments qui contiennent les nœuds correspondants ; on procède de la même façon pour construire le vecteur second

1 Le potentiel des efforts extérieurs ne dépend pas du déplacement de la structure, c'est ce qu'on appelle un chargement *mort* ou *non-suiveur*. Dans les cas des grandes déformations, les chargements de type pression ne peuvent pas respecter cette hypothèse.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

membre L_i. Ces contributions, appelées termes élémentaires, sont calculées lors d'une boucle sur les éléments et ne dépendent que des seules variables de l'élément Ω^e :

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

La relation entre le tenseur des contraintes de Cauchy $\boldsymbol{\sigma}$ et les déplacements $\mathbf{u}$ est donnée par la relation de comportement, et est indépendante de l'écriture de la formulation variationnelle. Dans le cas élastique, on a :

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

Λ_{ijkl} est le tenseur d'élasticité de Hooke. Cette forme tensorielle est peu pratique, on utilise préférentiellement la notation de Voigt, qui permet d'écrire :

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

En coordonnées cartésiennes, on a :

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

Et une forme modifiée des composantes de déformation pour permettre d'exprimer le produit contracté, soit :

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

Remarque importante :

Dans l'intégration des lois de comportement, les composantes de cisaillement des contraintes et de déformations utilisés par Code_Aster *sont :*

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

Le produit de ces deux vecteurs donne bien le même résultat que le double produit contracté (49).

Avec cette nouvelle notation, nous avons en élasticité :

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

Nous repartons de l'écriture EF du champ des déplacements :

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

Et, de manière analogue, le champ des déplacements virtuels :

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

Par souci de simplification des notations, on omettra la référence à l'élément. Il est nécessaire d'xprimer le tenseur des déformations (virtuelles ou réelles) :

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

On obtient alors pour la matrice relative à la forme bilinéaire :

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

Les matrices $\left[B\right]$ et $[\Lambda]$ contiennent la non-linéarité éventuelle du comportement et dépendront des déplacements :

1. $\left[B\right]$ est une fonction des déplacements si on se trouve dans la situation des grandes déformations ou des grandes transformations (grandes rotations et/ou grands déplacements).
2. $[\Lambda]$ est la matrice de comportement. Elle devient dépendante des déplacements (et d'autres variables) dans le cas des comportements non-linéaires et/ou non-élastiques.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Dans ces deux cas, le processus de résolution des équations impliquera un traitement spécifique (le plus souvent, une linéarisation de type Newton-Raphson). De manière analogue, on obtiendra facilement la forme élémentaire pour le second membre.

5.3 Calcul des termes élémentaires

Les termes élémentaires à calculer sont de la forme :

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

Trois types d'opérations sont à effectuer :

1. la transformation des dérivées par rapport à $\mathbf{x}$ en dérivées par rapport à $\boldsymbol{\xi}$;
2. le passage d'une intégration sur l'élément réel à une intégration sur l'élément de référence,
3. la réalisation numérique de cette intégration qui est généralement faite par une formule de quadrature.

5.3.1 Transformation des dérivées

La transformation des dérivées s'effectue grâce à la matrice jacobienne $\boldsymbol{J}$, d'après la règle de dérivation en chaîne :

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

où $\boldsymbol{u}_\alpha^{\mathrm{nod}}$ est le vecteur des valeurs nodales de la composante α du déplacement.

5.3.2 Changement de domaine d'intégration

Le passage à l'intégration sur l'élément de référence s'effectue en multipliant l'intégrande par le déterminant de la matrice jacobienne, appelé jacobien :

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

Le passage de l'élément de référence à l'élément réel implique la bijectivité de la transformation τ. Il faut donc $\det(\boldsymbol{J}(\boldsymbol{\xi}))\neq 0$, ce qui implique que l'élément ne doit pas se retourner ou dégénérer (par exemple il ne faut pas que le quadrangle dégénère en triangle).

5.3.3 Intégration numérique

Dans certains cas particuliers, on peut calculer analytiquement les intégrales. Par exemple, pour un triangle en deux dimensions, le Jacobien est constant sur le triangle, et les intégrandes se ramènent à des monômes que l'on sait intégrer exactement grâce à la formule d'intégration numérique dite « de Gauss »[2] :

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

Cependant, ces cas particuliers sont rares, et on préfère évaluer numériquement les intégrales en faisant appel à des formules de quadrature. Celles-ci donnent une approximation de l'intégrale sous forme d'une somme pondérée des valeurs de l'intégrande en un certain nombre de points de l'élément appelés points d'intégration :

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

2 Par abus de langage, on appelle fréquemment les schémas d'intégration numériques « schémas de Gauss » bien qu'il en existe de plusieurs sortes (Hammer pour le triangle, Gauss-Radau, Newton-Cotes, etc.).

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Les scalaires ω_g sont appelés les poids d'intégration, et les coordonnées ξ_g sont les coordonnées des r points d'intégration dans l'élément de référence.
Dans les méthodes d'intégration de Gauss, les points et poids d'intégration sont déterminés de manière à intégrer exactement des polynômes d'ordre donné. C'est ce type de méthode que l'on utilise dans *Code_Aster*, les points d'intégration s'appellent alors des points de Gauss.
Le nombre de points de Gauss choisi permet d'intégrer exactement dans l'élément de référence. En fait, à cause de la non-linéarité éventuelle de la transformation géométrique ou de la dépendance spatiale des coefficients (par exemple pour des éléments déformées ou de second ordre), l'intégration n'est pas exacte dans l'élément réel.
Pour chaque élément Ω_e, on a su calculer les termes dits élémentaires : matrice élémentaire $\mathbf{A}^e$ et vecteur élémentaire $\mathbf{L}^e$. La matrice A et le vecteur L sont obtenus par une procédure que l'on appelle l'assemblage des termes élémentaires.
Si on reprend la forme élémentaire de rigidité:

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

L'intégration numérique implique que l'on évalue les contraintes et les déformations aux points d'intégration:

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

Ce qui signifie que les contraintes et les déformations sont les plus exactes (ou les moins fausses) aux points d'intégration (champs dits « ELGA » dans *Code_Aster*). Le simple fait d'extrapoler ces valeurs aux nœuds pour l'affichage introduit une erreur. Il s'agit d'ailleurs d'une méthode d'évaluation de l'erreur, appelée indicateur d'erreur de Zhu-Zienkiewicz.
En élasticité 2D, un triangle exhibant un jacobien constant, un seul point de Gauss est suffisant pour intégrer exactement les termes de la matrice et du second membre (s'il est constant).
Le coût calcul augmente avec le nombre de points d'intégration, particulièrement pour les lois de comportement non-linéaires. Par exemple, un hexaèdre à 27 nœuds a besoin de 27 points de Gauss pour intégrer les quantités. Il arrive donc fréquemment qu'on « sous-intègre », c'est à dire qu'on utilise moins de points d'intégration que le minimum requis, commettant ainsi une erreur que l'on compensera éventuellement par un maillage plus fin. En plus de cette erreur systématique, cette sous-intégration doit être faite avec précaution car elle peut produire des défauts de rang de la matrice et rendre ainsi le système linéaire non inversible.

6 Résolution du système matriciel

On obtient donc un système linéaire à résoudre:

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

Quel que soit le champ des déplacements virtuels, donc:

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

6.1 Imposition des conditions aux limites cinématiques

Le traitement des conditions aux limites cinématiques du type $\mathbf{u}=\mathbf{u}^{\mathbf{D}}$ se fait de deux manières différentes:

1. La méthode « cinématique » (AFFE_CHAR_CINE dans Code_Aster) consiste à modifier la matrice et le second membre. Cette méthode est rapide et n'introduit pas de variables supplémentaires. Par contre, elle n'est pas générale et ne permet pas d'appliquer des conditions limites complexes du style $\sum \boldsymbol{u}_i.a_i=\boldsymbol{u}^{\boldsymbol{D}}$.
2. La méthode par dualisation (AFFE_CHAR_MECA dans Code_Aster) consiste à introduire un vecteur de multiplicateurs (ou paramètres) de Lagrange $\boldsymbol{\lambda}$, ce qui augmente le nombre d'inconnues mais permet de traiter tous les cas.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

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

6.2 Résolution

Le système linéaire peut être résolu par un certain nombre de méthodes numériques. Les méthodes utilisées dans *Code_Aster* sont une factorisation LDL^T par blocs, une méthode multi-frontale (ou son équivalent avec pivotage, MUMPS), un gradient conjugué préconditionné ainsi que la collection de solveurs itératifs PETSC.
Les méthodes de résolution se divisent en deux catégories :

- Les méthodes directes qui résolvent exactement (aux erreurs numériques près)
- Les méthodes itératives qui construisent une suite de vecteurs convergeant vers la solution

Les matrices issues de la méthode des éléments finis sont très creuses (elles comportent une majorité de termes nuls). En pratique, sur des systèmes de taille standard (quelques dizaines de milliers d'équations), la densité de termes non nuls dépasse rarement les 0.01%. Elles sont donc stockées sous forme creuse (ou « sparse ») et prennent peu de place en mémoire. A contrario, les matrices ne sont pas construites pour être utilisés efficacement avec les bibliothèques de programmes mathématiques optimisées dédiées aux matrices pleines (libraires BLAS par exemple). Des solveurs sont donc développés spécifiquement pour ces problèmes.
Un solveur direct a pour principe de décomposer la matrice en un produit de matrices de forme particulières. Par exemple, la décomposition LDL^T :

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

Où la matrice D est diagonale et la matrice L est triangulaire inférieure. Cette décomposition n'est valable que pour les matrices symétriques. Si ce n'est pas le cas, il faut utiliser d'autres décompositions.
Le principe est le suivant :

- A partir de la matrice initiale (très creuse), on construit un produit de matrices remarquables. C'est l'opération de *factorisation*.
- Ces matrices remarquables permettent de résoudre le problème de manière très rapide. C'est la phase de *descente-remontée.*

La phase de factorisation est la plus coûteuse. Pour les décompositions les plus répandues, le coût machine est en n^3 où n est le nombre d'équations. Le coût mémoire dépendra du profil de la matrice (de la numérotation des éléments finis). Des procédés automatiques cherchent à optimiser cette numérotation pour avoir une structure aussi compacte que possible. Même avec cette optimisation, il est fréquent que la matrice factorisée prennent plusieurs centaines de fois, voire plusieurs milliers de fois plus de mémoire que la matrice initiale. Les solveurs directs consomment donc beaucoup de mémoire et ça en devient rédhibitoire à partir de plusieurs centaines de milliers de degrés de liberté, même sur les machines les plus puissantes. Par contre, ces méthodes directes sont particulièrement robustes. Les problèmes en mécanique des structures et des solides conduisent bien souvent à des matrices avec un mauvais conditionnement (c'est particulièrement le cas de toutes les dernières innovations numériques qui utilisent des méthodes mixtes avec beaucoup de multiplicateurs de Lagrange).
Lorsque c'est possible, on utilise préférentiellement des méthodes itératives dont le principe consiste à trouver une approximation de l'inverse de la matrice et à procéder ensuite à une résolution itérative, pas par pas, qui n'utilise que des produits matrices-vecteurs, très efficaces et peu coûteux en mémoire.
Toutefois, ces méthodes itératives ont plusieurs défauts :

- Elles sont moins robustes que les méthodes directes, particulièrement lorsque le conditionnement est mauvais
- Les méthodes de pré-conditionnement sont très nombreuses et il en existe autant que de problèmes différents (voire plusieurs possibles par problème). Ce qui oblige l'utilisateur à jongler avec les différentes méthodes, sans jamais être assuré d'obtenir un résultat à la fin.
- Ce sont des méthodes itératives, ce qui implique un critère d'arrêt du processus, et donc un paramètre à gérer mais aussi des problèmes de cumul d'erreurs d'arrondi.

7 Organisation d'un calcul par éléments finis dans le *Code_Aster*

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

On décrit très brièvement comment et à quel endroit les aspects évoqués dans ce document sont implantés dans *Code_Aster*.

7.1 Notion d'élément fini dans *Code_Aster*

Un type d'élément fini est défini par :
- un type de maille
- une liste de nœuds
- des fonctions de forme
- des options de calcul

Un élément dans le maillage est défini par un type de maille, une géométrie (coordonnées des nœuds) et une topologie (liste ordonnée des nœuds). C'est le type de modélisation choisi dans le fichier de commande qui permet d'affecter à chaque maille du maillage un type d'élément fini. La commande AFFE_MODELE [U4.22.01] affecte à chaque maille un type d'élément fini correspondant à la modélisation spécifiée pour cette maille.

Remarque importante :

Il ne faut pas oublier d'affecter des éléments finis aux mailles de bord dont on a besoin pour imposer les conditions aux limites et chargements, et qu'on aura pris soin de créer lors de la fabrication du maillage.

L'opérateur AFFE_CHAR_MECA [U4.44.01], qui affecte conditions aux limites et chargements, va également créer des éléments finis, par exemple les éléments finis qui porteront les degrés de liberté de LAGRANGE utilisés dans la dualisation des conditions aux limites [R3.03.01].
L'opérateur AFFE_CARA_ELEM [U4.42.01] permet de définir des caractéristiques supplémentaires pour certains types d'éléments : par exemple, l'épaisseur des coques, l'orientation des poutres, les matrices de masse et de rigidité des éléments discrets.
Une option de calcul indique le type de calcul élémentaire que l'élément est capable de calculer. Par exemple RIGI_MECA concerne le calcul de la matrice élémentaire de rigidité mécanique :

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

Les « données » de cette option sont la géométrie Ω^e et le matériau Λ, complétées par la température si le matériau en dépend.
Rappelons que pour appliquer les chargements de frontière, on utilise des éléments finis de bord particuliers, et non pas les frontières des éléments finis de volume (3D) ou de surface (2D).

Remarque :

Un développeur peut parfois avoir le choix entre créer un nouvel élément fini ou ajouter une option de calcul à un élément existant ; le choix entre ces deux solutions tient en général compte de critères de facilité informatique (ex. éléments sous-intégrés).

7.2 Initialisations des éléments

L'utilisation d'éléments de référence permet d'effectuer un certain nombre de calculs une fois pour toutes au début de l'exécution. On définit, pour chaque type d'élément de référence :
- Le nombre de nœuds et leurs coordonnées ;
- Le nombre de familles de points de GAUSS ;
- Le nombre de points de GAUSS ;
- Les poids d'intégration ω_g ;
- Les valeurs des fonctions de forme aux points de Gauss $N_i(\boldsymbol{\xi}_\mathbf{g})$;
- Les valeurs des dérivées des fonctions de forme aux points de Gauss $\frac{\partial N_i(\boldsymbol{\xi}_\mathbf{g})}{\partial \boldsymbol{\xi}}$.

Pour un élément donné, on n'intègre pas forcément tous les termes élémentaires avec le même nombre de points de Gauss : par exemple, on utilise en général plus de points de Gauss pour la matrice de masse que pour la matrice de rigidité, car les produits de fonctions de forme sont de degré plus élevé que les produits de

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

leurs dérivées. Un autre exemple est la sous-intégration utilisée dans certains cas. On appelle famille de points de Gauss chaque ensemble de points de Gauss susceptible d'être utilisé.

7.3 Calcul des termes élémentaires

Lors du calcul des termes élémentaires (dans les routines TE....), on effectue pour chaque point de Gauss les opérations suivantes :

- Calcul des dérivées des fonctions de forme sur l'élément réel à partir des coordonnées des nœuds de l'élément et des dérivées des fonctions de forme sur l'élément de référence ;
- Calcul de la matrice jacobienne ;
- Récupération du poids d'intégration multiplié par le Jacobien au point de GAUSS considéré ;
- Évaluation de l'intégrande (selon l'option calculée).

Le terme élémentaire est calculé par somme sur les points de Gauss en pondérant par les poids d'intégration.

7.4 Résolution globale

La résolution globale a lieu dans les routines OP.... de haut niveau correspondant aux commandes utilisateur (MECA_STATIQUE [U4.51.01], STAT_NON_LINE [U4.51.03], THER_LINEAIRE [U4.54.01], etc.).

8 Bibliographie

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

2) A. Ern, J.-L. Guermond, « Éléments finis : théorie, applications, mise en oeuvre », Springer, 2002.

3) G. Dhatt, G. Touzot, E. Lefrançois, « Méthode des éléments finis : une présentation », Hermès, 2004.

9 Description des versions du document

Version Aster	Auteur(s) Organisme(s)	Description des modifications
3	I.VAUTIER	Texte initial
10.2	M.Abbas	Ré-écriture partielle, « anonymisation » des concepts par rapport à la mécanique

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)