

Calcul des structures par éléments finis

Antoine Legay
Cnam-Paris
2015-2016

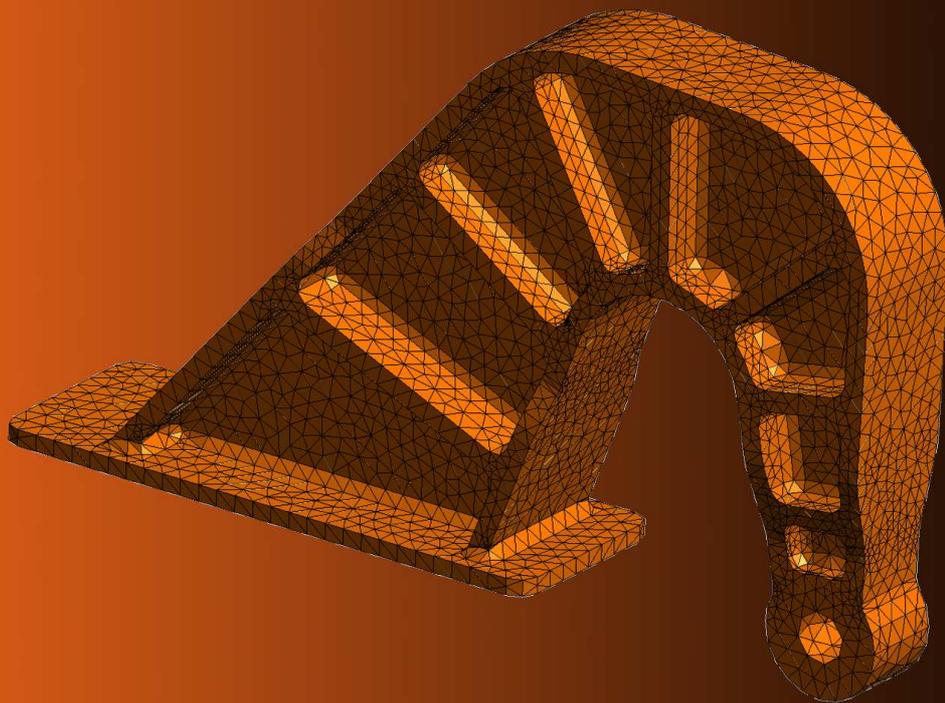


Table des matières

I Introduction	1
I.1 Outil d'aide au dimensionnement	1
I.2 Du réel au modèle mathématique mécanique	2
I.2.1 Problème réel	2
I.2.2 Modèle mathématique mécanique	2
I.3 Espace vectoriel associé au modèle mathématique mécanique	3
I.3.1 Ensemble des champs de déplacements	3
I.3.2 Espace vectoriel des champs de déplacements	4
I.3.3 Définitions et propriétés d'un espace vectoriel	5
I.4 Vers le modèle éléments finis	7
I.5 Les sources d'erreurs	7
I.5.1 Erreur entre le problème réel et le modèle mathématique	7
I.5.2 Erreur entre le modèle mathématique et le modèle éléments finis	8
I.6 Élément fini de barre pour le calcul des treillis	8
I.6.1 Introduction à l'élément de barre	8
I.6.2 Équilibre de l'élément barre	9
I.6.3 Étude de deux barres	10
I.6.4 Élément barre pour le calcul des treillis plans	13
II Problème de référence	15
II.1 Rappel de mécanique des milieux continus	15
II.2 Champ de déplacement cinématiquement admissible	17
II.2.1 Espace vectoriel des champs de déplacements	17
II.2.2 Espace des champs de déplacements cinématiquement admissibles	17

II.3	Champ de contrainte statiquement admissible	17
II.3.1	Espace vectoriel des champs de contraintes	17
II.3.2	Espace des champs de contraintes statiquement admissibles	19
II.4	Écriture du problème de référence	20
III	Écriture variationnelle	21
III.1	Écriture variationnelle de l'ensemble des champs de contraintes admissibles	21
III.2	Formulation variationnelle en déplacement	24
III.2.1	Loi de comportement en notations de Voigt	24
III.2.2	Opérateur gradient symétrisé en notations de Voigt	24
III.2.3	Formulation variationnelle en déplacement, écriture en notations de Voigt	25
III.3	Équivalence énergétique	26
III.4	Formes bilinéaire et linéaire, norme en énergie	27
III.4.1	Forme linéaire	27
III.4.2	Forme bilinéaire	28
III.4.3	Écriture du problème de référence	28
III.4.4	Norme en énergie	28
IV	Espace d'approximation E.F.	29
IV.1	Introduction	29
IV.2	Fonction à une variable	29
IV.3	Fonction à deux variables	32
IV.4	Fonction vectorielle à deux variables	35
IV.5	Fonction à trois variables	35
IV.6	Fonction vectorielle à trois variables	37
V	Construction de la base E.F.	39
V.1	Construction des fonctions de forme dans un élément	39
V.1.1	Introduction	39
V.1.2	Fonctions de forme dans un élément unidimensionnel	39
V.1.3	Fonctions de forme dans un élément du plan	41
V.1.4	Fonctions de forme dans un élément tridimensionnel	44
V.2	Table de connectivité et coordonnées des nœuds	45
VI	Problème discrétisé	47
VI.1	Définition de l'erreur de discrétisation	47
VI.2	Écriture du problème discrétisé	47
VI.2.1	Rappel des différents espaces introduits	47
VI.2.2	Solution en déplacement	48
VI.2.3	Écriture matricielle du problème discrétisé	49
VI.3	Résolution du problème discrétisé	50
VI.3.1	Séparation des équations	50
VI.3.2	Résolution en déplacements	51
VI.3.3	Réactions aux appuis	51
VI.3.4	Système matriciel complet	52
VI.3.5	Blocage des mouvements de solides rigides	52

VI.4 Construction du problème discrétisé 53

VI.4.1 Calcul de la matrice de rigidité 53

VI.4.2 Calcul des forces aux nœuds 55

VII Calculs au niveau élémentaire 57

VII.1 Élément de référence 57

VII.1.1 Nécessité d'un élément de référence 57

VII.1.2 Géométrie de l'élément de référence 58

VII.1.3 Fonctions de forme un élément de référence à 2 variables 58

VII.1.4 Passage de l'élément de référence à l'élément physique 61

VII.2 Matrice de rigidité dans l'élément de référence 63

VII.2.1 Expression 63

VII.2.2 Cas du quadrangle 63

VIII Intégration numérique. 65

VIII.1 Intégration numérique de la matrice de rigidité 65

VIII.2 Intégration des fonctions à une variable 66

VIII.2.1 Polynômes de Lagrange 66

VIII.2.2 Méthode de Newton-Côtes 68

VIII.2.3 Méthode de Gauss 69

VIII.3 Intégration des fonctions à deux variables 70

VIII.3.1 Cas des éléments quadrangles 70

VIII.3.2 Cas des éléments triangles 72

VIII.4 Intégration des fonctions à trois variables 73

VIII.4.1 Cas des éléments hexaèdres 73

VIII.4.2 Cas des éléments tétraèdres 74

VIII.5 Critères de qualité du maillage 74

IX Analyse des résultats E.F. 77

IX.1 Post-traitement, lissage des contraintes 77

IX.1.1 Propriétés de la solution éléments finis 77

IX.1.2 Contrainte lissée 77

IX.1.3 Implémentation et résolution 79

IX.1.4 Application au triangle à 3 nœuds 80

IX.2 Estimateur d'erreur 81

IX.2.1 Rappel sur l'erreur en déplacement 81

IX.2.2 Estimateur à priori 81

IX.2.3 Estimateur à posteriori 82

IX.2.4 Maillage adaptatif 84

X Éléments finis de barre et poutre 85

X.1 Modèle de poutre 85

X.1.1 Modèle géométrique 85

X.1.2 Hypothèses cinématiques 86

X.1.3 Écriture de la relation déformations-déplacement 86

X.1.4 Relation de comportement 87

X.1.5 Énergie de déformation 87

X.1.6 Hypothèses sur les déformations transverses 88

X.2 Élément fini de traction-compression : élément barre 89

- X.2.1 Espace discrétisé 89
- X.2.2 Opérateur gradient discrétisé 89
- X.2.3 Matrice de rigidité 89
- X.2.4 Calcul de treillis de barres 90

X.3 Élément fini de flexion de type Euler-Bernoulli 92

- X.3.1 Espace discrétisé 92
- X.3.2 Opérateur gradient discrétisé 94
- X.3.3 Matrice de rigidité 94

XI Élément fini de plaque 95

XI.1 Modèle de plaque 95

- XI.1.1 Modèle géométrique 95
- XI.1.2 Hypothèse cinématique 96
- XI.1.3 Écriture de la relation déformations-déplacements 97
- XI.1.4 Relation de comportement entre les déformations et les contraintes 97
- XI.1.5 Énergie de déformation 98
- XI.1.6 Hypothèses sur les déformations transverses 99

XI.2 Élément fini de type Reissner-Mindlin 100

- XI.2.1 Espace discrétisé 100
- XI.2.2 Opérateur gradient discrétisé 100
- XI.2.3 Matrice de rigidité 102
- XI.2.4 Forces extérieures généralisées 102

XI.3 Assemblage de plaques dans l'espace 103

XII Décomposition de domaine. 105

XII.1 Objectifs 105

XII.2 Méthode de Schur primale 105

- XII.2.1 Principe 105
- XII.2.2 Stratégie de résolution globale 108
- XII.2.3 Généralisation à plusieurs sous-structures 108
- XII.2.4 Calcul du complément de Schur 109

XII.3 Méthode de Schur duale 111

- XII.3.1 Principe 111
- XII.3.2 Discrétisation 111
- XII.3.3 Couplage de maillages incompatibles 112

XIII Initiation à la programmation 113

XIII.1 Langage de programmation 113

XIII.2 Le langage Python 114

- XIII.2.1 Les bases du langage Python et son installation 114
- XIII.2.2 Les conditions et boucles en Python 115
- XIII.2.3 Les fonctions en Python 117
- XIII.2.4 Le calcul matriciel en Python 118

XIII.3 Le langage Fortran 120

- XIII.3.1 Les bases du langage Fortran et son installation 120
- XIII.3.2 Les conditions et boucles en Fortran 122
- XIII.3.3 Les fonctions en Fortran 122
- XIII.3.4 Le calcul matriciel en Fortran 126

XIII.4 Utilisation de routines Fortran dans un programme en Python 126

XIV Code SILEX 129

XIV.1 Présentation 129

XIV.2 Cœur écrit en Fortran de SILEX pour le tétraèdre à 4 nœuds 130

XIV.2.1 Matrice de rigidité élémentaire 130

XIV.2.2 Matrice de rigidité globale 131

XIV.3 Programme principal en Python 138

XIV.4 Gmsh : un logiciel libre de maillage et de visualisation 144

XIV.5 Un exemple de calcul mené avec SILEX 144

XIV.5.1 Géométrie et conditions aux limites 144

XIV.5.2 Analyse des résultats 144

XIV.5.3 Conclusions de l'étude 145

I — Introduction

I.1 Outil d'aide au dimensionnement

Le but du dimensionnement des structures est de déterminer les formes, dimensions, matériaux afin de satisfaire la fonction demandée.

On peut distinguer deux grands types de chargement sur une structure :

1. chargement en fonctionnement normal,
2. chargement en situation extrême.

Pour les situations extrêmes, l'étude est souvent dynamique en non-linéaire. Les critères de dimensionnement sont souvent liés à la sécurité, par exemple :

- les passagers d'un véhicule automobile sont-ils bien protégés en cas de choc ?
- suite à un seisme et en fonction de son intensité, combien de temps un barrage va t-il tenir ?

Concernant les chargements en fonctionnement normal (étude linéaire le plus souvent), on peut répertorier les cas suivants :

- Statique :
 - Les contraintes sont-elles dans le domaine élastique ?
 - Les déplacements sont-ils acceptables ?
 - Y-a-t-il un risque de flambage ?
- Dynamique, analyse modale :
 - Les fréquences propres sont-elles proches des fréquences d'excitation ?
 - Les formes des modes sont-elles acceptables par rapport à l'utilisation ?



Figure I.1 – Problème réel : dimensionnement des pieds d'une table.

- Dynamique, analyse fréquentielle :
 - Comment la structure répond à une sollicitation harmonique ?
 - Que valent les amortissements ?
 - Quelle est la durée de vie de la structure en fatigue ?
- Dynamique, transitoire, domaine temporel :
 - Comment la structure répond à un choc ?

La méthode des éléments finis peut être utilisée dans tous ces cas comme aide au dimensionnement.

I.2 Du réel au modèle mathématique mécanique

I.2.1 Problème réel

On se place dans le cadre d'un problème de statique, élastique et linéaire. Le problème réel fait intervenir (Fig. I.1) :

- Une structure, comprenant des incertitudes sur sa géométrie et son matériau ;
- Des liaisons avec l'extérieur, souvent assez mal maîtrisées ;
- Des efforts appliqués, parfois assez complexes.

Lors de la phase de conception, la solution de ce problème n'est pas accessible. Une fois la structure fabriquée et placée dans son environnement, la solution est partiellement accessible par des mesures (jauge de déformation, photoélasticité, ...).

I.2.2 Modèle mathématique mécanique

Afin de trouver une solution approchée mathématique du problème réel, on utilise un modèle mathématique du problème réel. Les modèles généralement utilisés en mécanique sont :

- le modèle de poutre,
- le modèle de plaque,
- le modèle de coque,

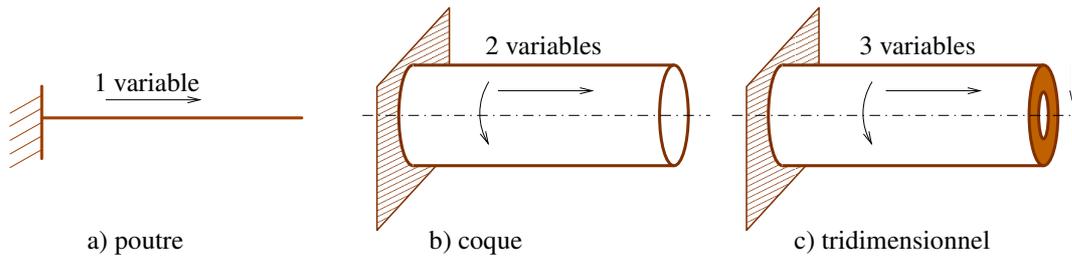


Figure 1.2 – Trois modèles du pied de table.

- le modèle plan en contraintes planes,
- le modèle plan en déformations planes,
- le modèle axisymétrique,
- le modèle tri-dimensionnel.

Pour l'exemple précédent d'un pied de table, on peut par exemple choisir :

- Le modèle de poutre (Fig. I.2 a) :
 - hypothèse cinématique de poutre
 - 1 variable le long de l'axe de la poutre décrit le problème
 - encastrement de type poutre
 - torseurs d'efforts équivalents
- Le modèle de coque (Fig. I.2 b) :
 - hypothèse cinématique de coque
 - 2 variables sur la surface moyenne de la coque décrivent le problème
 - encastrement de type coque
 - torseurs d'efforts équivalents distribués
- Le modèle tri-dimensionnel (Fig. I.2 c) :
 - encastrement tri-dimensionnel
 - 3 variables dans les 3 directions de l'espace décrivent le problème
 - forces surfaciques distribuées

Pour les trois modèles proposés, l'encastrement est modélisé de façon parfaite alors que la liaison réelle est réalisée par une pièce intermédiaire souple. Ces modèles ne permettent pas de dimensionner cette pièce intermédiaire. C'est au concepteur de choisir le modèle le plus adapté par rapport aux critères de dimensionnement qu'il pense être les plus judicieux.

I.3 Espace vectoriel associé au modèle mathématique mécanique

I.3.1 Ensemble des champs de déplacements

Quelque-soit le modèle mathématique mécanique choisi, on cherche un champ de déplacement \vec{u} défini sur le domaine idéalisé Ω . On se place dans la suite dans le cas d'un modèle tridimensionnel,

le champ de déplacement \vec{u} est défini comme :

$$\begin{aligned} \vec{u} \text{ "régulier"} : \Omega \subset \mathbb{R}^3 &\rightarrow \mathbb{R}^3 \\ M &\mapsto \vec{v}(M) \end{aligned}$$

On pose l'ensemble \mathcal{V} des champs de déplacement "suffisamment réguliers" définis de Ω vers \mathbb{R}^3 :

$$\mathcal{V} = \left\{ \begin{array}{l} \vec{v} \text{ "régulier"} : \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3 \\ M \mapsto \vec{v}(M) \end{array} \right\}$$

La notion de fonction "suffisamment régulière" n'est pas détaillée ici, cela fait appel à des propriétés mathématiques des fonctions qui permettent de les dériver, de les intégrer, et plus particulièrement d'intégrer le carré de leurs dérivées. Ces espaces sont appelés "espace de Sobolev" ou "espace de Hilbert", il est nécessaire de se placer dans ces espaces pour que les développements effectués dans ce cours soient corrects d'un point de vue mathématique. Cependant, ce cours ayant pour vocation d'expliquer les choses simplement, nous parlerons de fonction "suffisamment régulière" dans la suite.

I.3.2 Espace vectoriel des champs de déplacements

L'ensemble \mathcal{V} muni d'un loi interne, notée $+$, et d'une loi externe avec \mathbb{R} , notée \times , est un espace vectoriel sur \mathbb{R} de dimension infinie. Un élément de cet espace vectoriel est un champ de vecteurs de \mathbb{R}^3 .

L'opération interne $+$ d'addition dans \mathcal{V} est définie comme :

$$\forall \vec{u}, \vec{v} \in \mathcal{V}, \vec{w} = \vec{u} + \vec{v}$$

où $\vec{w} \in \mathcal{V}$ vaut

$$\forall M \in \Omega, \vec{w}(M) = \vec{u}(M) + \vec{v}(M).$$

L'opération externe \times de multiplication avec \mathbb{R} est définie comme :

$$\forall \vec{u} \in \mathcal{V} \text{ et } a \in \mathbb{R}, \vec{w} = a \times \vec{u}$$

où $\vec{w} \in \mathcal{V}$ vaut

$$\forall M \in \Omega, \vec{w}(M) = a \vec{u}(M).$$

L'élément neutre $\vec{0}_{\mathcal{V}}$ de \mathcal{V} est le champ de vecteurs qui a tout point $M \in \Omega$ associe le vecteur nul de \mathbb{R}^3 .

Preuve On vérifie aisément que :

$$\forall \vec{u}, \vec{v}, \vec{w} \in \mathcal{V}$$

et $\forall a, b \in \mathbb{R}$, on a :

$$\begin{aligned} \vec{u} + \vec{v} &= \vec{v} + \vec{u} \in \mathcal{V} \\ \vec{u} + (\vec{v} + \vec{w}) &= (\vec{u} + \vec{v}) + \vec{w} \\ \vec{u} + (-\vec{u}) &= \vec{0}_{\mathcal{V}} \quad (-\vec{u} \text{ désigne l'opposé de } \vec{u}) \\ \vec{u} + \vec{0}_{\mathcal{V}} &= \vec{u} \\ a \times (\vec{u} + \vec{v}) &= a \times \vec{u} + a \times \vec{v} \\ (a + b) \times \vec{u} &= a \times \vec{u} + b \times \vec{u} \\ (ab) \times \vec{u} &= a \times (b \times \vec{u}) \\ 1 \times \vec{u} &= \vec{u} \end{aligned}$$

I.3.3 Définitions et propriétés d'un espace vectoriel

Combinaison linéaire d'une famille de vecteurs

Soit une famille \mathcal{F} de n vecteurs de \mathcal{V} :

$$\mathcal{F} = (\vec{u}_1, \vec{u}_2, \dots, \vec{u}_n)$$

Tous vecteur $\vec{v} \in \mathcal{V}$ s'écrivant sous la forme

$$a_1 \times \vec{u}_1 + a_2 \times \vec{u}_2 + \dots + a_n \times \vec{u}_n$$

où $a_1, a_2, \dots, a_n \in \mathbb{R}$ est dit *combinaison linéaire* de la famille \mathcal{F} .

Famille libre et famille liée d'un espace vectoriel

Une famille est dite *libre* si chaque vecteur de la famille ne peut pas s'écrire comme une combinaison linéaire des autres vecteurs de la famille. Si une famille n'est pas libre, elle est *liée*, c'est à dire que au moins un vecteur de la famille peut s'écrire comme une combinaison linéaire des autres vecteurs de la famille.

Base et dimension d'un espace vectoriel

Une *base* est une famille libre qui permet d'écrire n'importe quel vecteur de l'espace vectoriel comme une combinaison linéaire des vecteurs de la base. Il existe une infinité de bases pour un espace vectoriel, mais elles ont toutes le même nombre de vecteurs.

La dimension d'un espace vectoriel est le nombre de vecteurs d'une base de cet espace vectoriel.

Si les bases n'ont pas un nombre fini de vecteurs, l'espace vectoriel est de dimension infinie.

Exercice I.1 Soit l'ensemble $M_{2,2}(\mathbb{R})$ des matrices à coefficients dans \mathbb{R} à 2 lignes et 2 colonnes, muni de l'addition $+$ des matrices et de la multiplication \times par un scalaire de \mathbb{R} des matrices.

1. Justifier que cet espace est un espace vectoriel.
2. Donner une famille libre, puis une famille liée.
3. Donner la base canonique, en déduire la dimension de cet espace vectoriel.

On donne la famille \mathbf{b} suivante :

$$\mathbf{b} = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} ; \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} ; \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} ; \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \right)$$

4. Écrire cette famille dans la base canonique.
5. Montrer que cette famille est une base.

On donne l'élément de $M_{2,2}(\mathbb{R})$ suivant :

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

6. Écrire A dans la base canonique puis dans la base \mathbf{b} .

Solution :

Cet espace est un espace vectoriel car :

$$\forall A, B, C \in M_{2,2}(\mathbb{R}); \forall a, b \in \mathbb{R}$$

- $A+B = B+A \in M_{2,2}(\mathbb{R})$
- $A+(B+C) = (A+B)+C$
- $A+(-A) = \mathbf{0}$ où $-A$ désigne l'opposé de A et $\mathbf{0}$ est l'élément neutre défini par :

$$\mathbf{0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

- $a \times (A+B) = a \times A + a \times B$
- $(a+b) \times A = a \times A + b \times A$
- $(ab) \times A = a \times (b \times A)$
- $1 \times A = A$

Un exemple de famille libre :

$$\left(\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} ; \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \right)$$

Un exemple de famille liée :

$$\left(\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} ; \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} ; \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \right)$$

car la troisième matrice est la somme des deux premières.

La base canonique est

$$e = \left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} ; \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} ; \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} ; \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \right)$$

Cette base contient 4 vecteurs, cet espace vectoriel est de dimension 4.

La famille b s'écrit dans la base canonique sous la forme :

$$b = \left(\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}_{(e)} ; \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}_{(e)} ; \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}_{(e)} ; \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}_{(e)} \right)$$

La matrice A s'écrit dans la base canonique comme :

$$A = 1 \times \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + 4 \times \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} + 2 \times \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + 3 \times \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

soit aussi

$$A = \begin{bmatrix} 1 \\ 4 \\ 2 \\ 3 \end{bmatrix}_{(e)}$$

Pour écrire A dans la base b , on cherche à écrire A comme une combinaison linéaire des matrices (ou vecteurs) de la base b :

$$\begin{bmatrix} 1 \\ 4 \\ 2 \\ 3 \end{bmatrix}_{(e)} = \alpha_1 \times \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}_{(e)} + \alpha_2 \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}_{(e)} + \alpha_3 \times \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}_{(e)} + \alpha_4 \times \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}_{(e)}$$

Soit le système linéaire d'équations suivant à résoudre :

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 2 \\ 3 \end{bmatrix}$$

La solution est :

$$\alpha_1 = 3 ; \alpha_2 = 3 ; \alpha_3 = 1 ; \alpha_4 = -2$$

Donc A dans la base b s'écrit

$$A = \begin{bmatrix} 3 \\ 3 \\ 1 \\ -2 \end{bmatrix}_{(b)}$$

I.4 Vers le modèle éléments finis

La démarche de dimensionnement consiste dans un premier temps à construire un modèle mathématique (ou idéalisé) du problème réel possédant un contenu mécanique que le concepteur doit choisir. Ce modèle mathématique englobe des hypothèses sur le problème réel :

- la géométrie,
- les liaisons,
- le chargement,
- le matériau,
- les équations que doivent satisfaire les variables du problème (en particulier le champ de déplacement)

Le champ de déplacement \vec{u} solution de ce problème mathématique appartient à un espace vectoriel noté \mathcal{V} de dimension infinie. Cet espace est trop “grand” pour y trouver la solution exacte du modèle mathématique, sauf dans quelques cas particuliers.

Dans un deuxième temps, l’idée de la méthode des éléments finis est de chercher une solution approchée du modèle mathématique dans un sous-espace vectoriel, noté \mathcal{V}^h , de \mathcal{V} et de dimension finie dont on se donne une base :

$$(\vec{\varphi}_1, \vec{\varphi}_2, \dots, \vec{\varphi}_n)$$

où chaque vecteur $\vec{\varphi}_i$ appartient à \mathcal{V} . La dimension de \mathcal{V}^h est le nombre de vecteurs que l’on se donne, ici n .

Tout vecteur de \mathcal{V}^h s’écrit alors comme une combinaison linéaire des vecteurs de la base :

$$a_1 \times \vec{\varphi}_1 + a_2 \times \vec{\varphi}_2 + \dots + a_n \times \vec{\varphi}_n$$

où $a_1, a_2, \dots, a_n \in \mathbb{R}$. La solution du modèle éléments finis \vec{u}^h est la combinaison linéaire qui se rapproche “au mieux” de la solution exacte du modèle mathématique.

I.5 Les sources d’erreurs

I.5.1 Erreur entre le problème réel et le modèle mathématique

L’erreur entre la solution réelle et la solution du modèle mathématique est appelée *l’erreur de modèle* (Fig. I.3).

Quantifier cette erreur nécessite de connaître la solution réelle :

- Lorsque la structure est construite, puis placée dans son environnement, cette solution est partiellement accessible par des mesures (jauges de déformations, matériau photélastique,...);
- Lorsque la structure est en phase de conception, la solution réelle est inaccessible.

Dans tous les cas, l’erreur de modèle est difficilement quantifiable, elle doit pourtant être maîtrisée et justifiée au mieux par le concepteur.

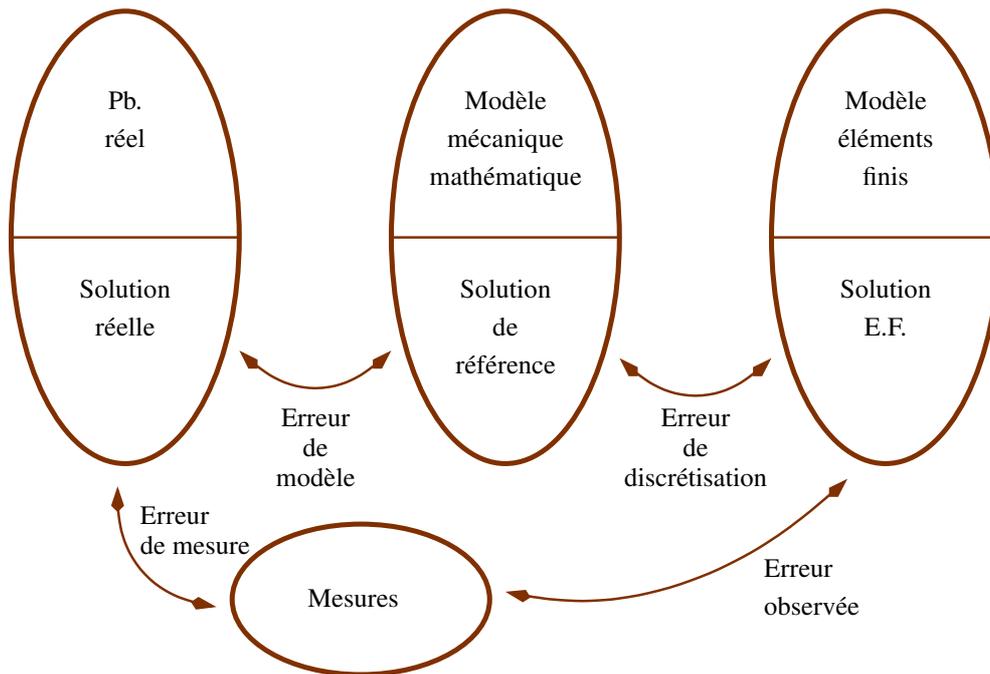


Figure I.3 – Liens entre le réel, le modèle mécanique et le modèle éléments finis.

I.5.2 Erreur entre le modèle mathématique et le modèle éléments finis

L'erreur entre la solution du modèle mathématique et la solution du modèle éléments finis est appelée *l'erreur de discrétisation* (Fig. I.3). Dans le cas d'un calcul de statique linéaire, cette erreur peut être estimée ; elle doit donc être parfaitement maîtrisée par le concepteur. Les ordinateurs actuels permettent de faire tendre cette erreur vers une erreur largement inférieure à celle de l'erreur de modèle. La tendance actuelle des logiciels de conception est d'intégrer un outil de calcul par éléments finis automatique assurant d'avoir un niveau d'erreur de discrétisation suffisamment faible pour confondre la solution mathématique et la solution éléments finis.

I.6 Élément fini de barre pour le calcul des treillis

I.6.1 Introduction à l'élément de barre

On désigne par barre une poutre travaillant seulement en traction-compression. Typiquement, les treillis de poutres sont souvent approximés dans un premier temps comme un ensemble de barres rotulées entre elles (Fig. I.4). Les 2 extrémités sont les nœuds de l'élément.

On présente dans cette partie l'élément barre de façon simple en utilisant les connaissances de mécanique d'un élève de Bac+2 :

- principe fondamental de la statique,
- torseur de cohésion,
- sollicitation de traction-compression pour une poutre,
- méthode des nœuds.

Le but est de mettre en œuvre rapidement la méthode. Cependant, la méthode des éléments finis ne peut pas se réduire à cette démarche simple dans le cas général d'une pièce tridimensionnelle.

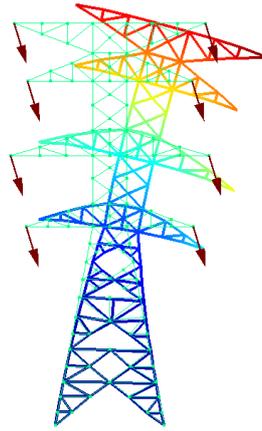


Figure I.4 – Exemple de calcul par éléments finis d'un pylône électrique (éléments barres).

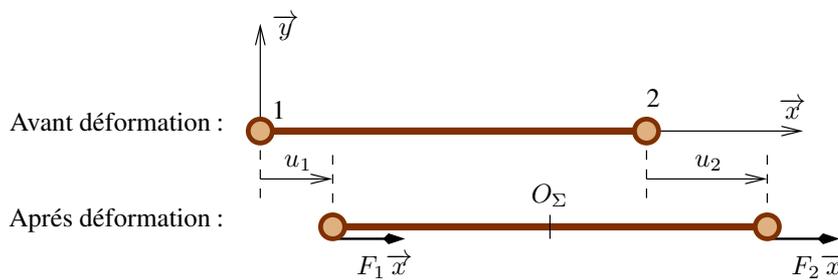


Figure I.5 – Élément barre à 2 noeuds.

Le même élément de barre est traité de façon plus cohérente avec le cours dans le chapitre X.

I.6.2 Équilibre de l'élément barre

Un élément de barre est représenté par un segment de droite reliant les deux extrémités de la barre (Fig. I.5). La barre est caractérisée par sa longueur l , par l'aire de sa section S et par son module d'Young E . Les extrémités sont appelés les nœuds de la barre, ils ont pour abscisses $x_1 = 0$ et $x_2 = l$.

La barre est en équilibre sous les actions des forces $F_1 \vec{x}$ et $F_2 \vec{x}$ aux nœuds 1 et 2. On note les déplacements des nœuds 1 et 2 respectivement $u_1 \vec{x}$, $u_2 \vec{x}$.

— Le torseur de cohésion en O_Σ peut s'écrire de deux façons :

$$\{\mathcal{K}_{O_\Sigma}\} = \{\text{ext. } \rightarrow 2\} \Rightarrow N = F_2$$

ou bien

$$\{\mathcal{K}_{O_\Sigma}\} = -\{\text{ext. } \rightarrow 1\} \Rightarrow N = -F_1$$

On remarque que la barre est bien en équilibre car $F_1 + F_2 = 0$.

— La relation entre N , u_1 et u_2 pour une poutre en traction est

$$N = \frac{ES}{l} \delta l \Rightarrow N = \frac{ES}{l} (u_2 - u_1)$$

— En utilisant les résultats précédents on a alors

$$F_1 = \frac{ES}{l} (u_1 - u_2)$$

$$F_2 = \frac{ES}{l}(-u_1 + u_2)$$

— Ces deux dernières relations s'écrivent sous la forme matricielle suivante :

$$\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \frac{ES}{l} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

La matrice

$$\mathbf{k} = \frac{ES}{l} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

est appelée la matrice de rigidité de l'élément barre.

Exercice I.2 On prend une poutre encastée à gauche ($u_1 = 0$), et on applique un effort F à l'extrémité droite ($F_2 = F$). Le système matriciel à résoudre est le suivant :

$$\frac{ES}{l} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$$

qui devient ici

$$\frac{ES}{l} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ u_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ F \end{bmatrix}$$

où les inconnues sont le déplacement de l'extrémité droite u_2 et la force de réaction de l'extrémité gauche F_1 . Ce système de 2 équations à 2 inconnues donne la solution

$$u_2 = \frac{Fl}{ES}$$

et

$$F_1 = -F$$

L'effort normal est déterminé par

$$N = \frac{ES}{l}(u_2 - u_1) = \frac{ES}{l}\left(\frac{Fl}{ES} - 0\right) = F$$

1.6.3 Étude de deux barres

Assemblage des matrices de rigidité élémentaires

On suppose deux barres de longueurs, de modules d'Young et de sections différentes collées bout à bout et soumises à de la traction (Fig. I.6). Les barres sont numérotées I et II, elles sont reliées à trois nœuds 1, 2 et 3. Ces trois nœuds subissent les forces extérieures $F_1 \vec{x}$, $F_2 \vec{x}$ et $F_3 \vec{x}$. L'équilibre global s'écrit

$$F_1 + F_2 + F_3 = 0.$$

A l'équilibre, l'ensemble des deux barres s'est déformé, les nœuds 1, 2 et 3 se sont déplacés respectivement de $u_1 \vec{x}$, $u_2 \vec{x}$ et $u_3 \vec{x}$ ($u_1 < u_2 < u_3$ si les deux barres sont en traction).

Les équilibres des nœuds s'écrivent :

— équilibre du nœud 1 (méthode des nœuds)

$$F_1 + N_1 = 0 \Rightarrow -N_1 = F_1$$

— équilibre du nœud 2

$$F_2 - N_1 + N_2 = 0 \Rightarrow N_1 - N_2 = F_2$$

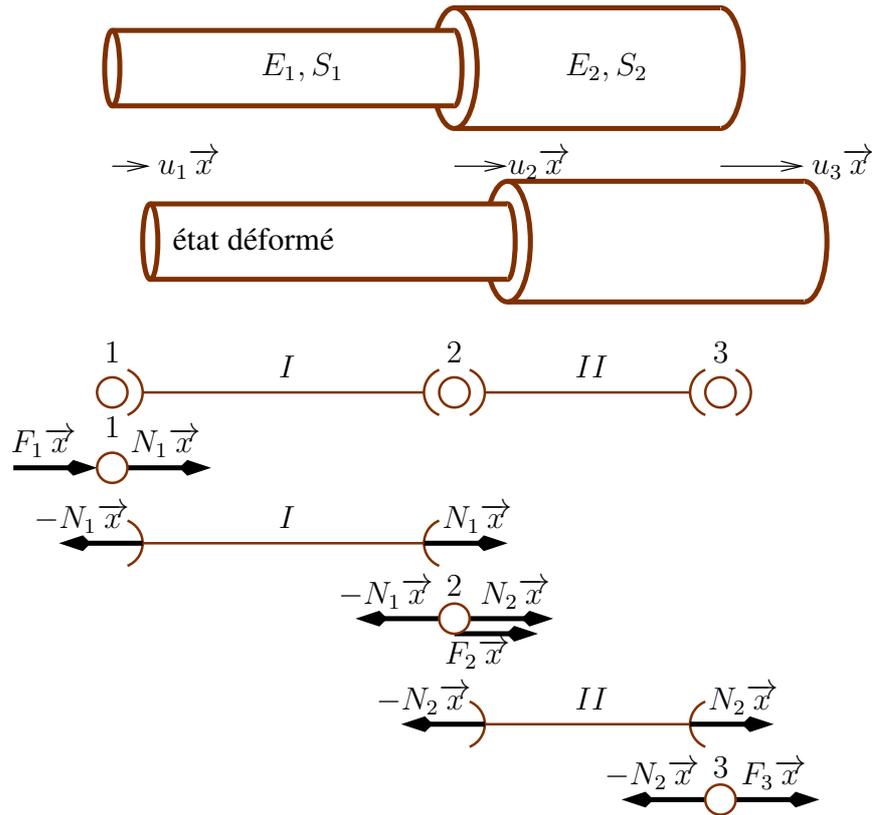


Figure I.6 – Deux barres en traction.

— équilibre du nœud 3

$$F_3 - N_2 = 0 \Rightarrow N_2 = F_3$$

Pour chaque élément, on peut relier l'effort normal à l'allongement de la barre :

— élément I :

$$N_1 = \frac{E_1 S_1}{L_1} (u_2 - u_1)$$

— élément II :

$$N_2 = \frac{E_2 S_2}{L_2} (u_3 - u_2)$$

En remplaçant dans les expressions des équilibres des nœuds, on a les trois équations suivantes :

$$\begin{aligned} -\frac{E_1 S_1}{L_1} (u_2 - u_1) &= F_1 \\ \frac{E_1 S_1}{L_1} (u_2 - u_1) - \frac{E_2 S_2}{L_2} (u_3 - u_2) &= F_2 \\ \frac{E_2 S_2}{L_2} (u_3 - u_2) &= F_3 \end{aligned}$$

En notant $k_1 = E_1 S_1 / L_1$ et $k_2 = E_2 S_2 / L_2$, on a alors le système matriciel suivant :

$$\underbrace{\begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_2 + k_1 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}}_{\mathbf{Q}} = \underbrace{\begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}}_{\mathbf{F}}$$

Cette opération est l'opération d'assemblage des matrices de rigidité élémentaires, la matrice \mathbf{K} est appelée matrice de rigidité de la structure, le vecteur \underline{Q} est le vecteur des inconnues de déplacements et le vecteur \underline{F} est le vecteur des forces extérieures :

$$\mathbf{K}\underline{Q} = \underline{F}$$

Mise en œuvre pratique

La première étape consiste à écrire les deux matrices de rigidité des deux éléments en repérant les lignes et les colonnes de chaque matrice par les inconnues de déplacements associées :

$$\mathbf{k}_1 = \begin{array}{cc} & \begin{array}{c} u_1 \\ u_2 \end{array} \\ \begin{array}{c} u_1 \\ u_2 \end{array} & \begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 \end{bmatrix} \end{array} \quad \text{et} \quad \mathbf{k}_2 = \begin{array}{cc} & \begin{array}{c} u_2 \\ u_3 \end{array} \\ \begin{array}{c} u_2 \\ u_3 \end{array} & \begin{bmatrix} k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \end{array} .$$

On range ensuite dans la matrice de rigidité \mathbf{K} de la structure chaque terme des deux matrices à la ligne et la colonne correspondante :

$$\mathbf{K} = \begin{array}{ccc} & \begin{array}{c} u_1 \\ u_2 \\ u_3 \end{array} \\ \begin{array}{c} u_1 \\ u_2 \\ u_3 \end{array} & \begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 + k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix} \end{array} .$$

Le système à résoudre est alors $\mathbf{K}\underline{Q} = \underline{F}$.

La deuxième étape consiste à faire le bilan des déplacements et des forces connus et inconnus. En prenant un encastrement à l'extrémité gauche et en appliquant une force F à l'extrémité droite, on a :

$$\underline{Q} = \begin{array}{c} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \\ \begin{array}{l} = 0, \text{ connu} \\ \text{inconnu} \\ \text{inconnu} \end{array} \end{array} \quad \underline{F} = \begin{array}{c} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \\ \begin{array}{l} \text{inconnu, réaction à l'encastrement} \\ = 0, \text{ connu} \\ = F, \text{ connu} \end{array} \end{array}$$

Si le déplacement est connu en un nœud alors la force est inconnue, si la force est connue alors le déplacement est inconnu.

La troisième étape est la résolution du système d'équations complet afin de déterminer toutes les inconnues

$$\begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_2 + k_1 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix} \begin{bmatrix} 0 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} F_1 \\ 0 \\ F \end{bmatrix}$$

Une fois toutes les inconnues trouvées, on peut calculer les efforts normaux dans chaque barre :

$$N_1 = \frac{E_1 S_1}{l_1} (u_2 - \underbrace{u_1}_{=0})$$

et

$$N_2 = \frac{E_2 S_2}{l_2} (u_3 - u_2).$$

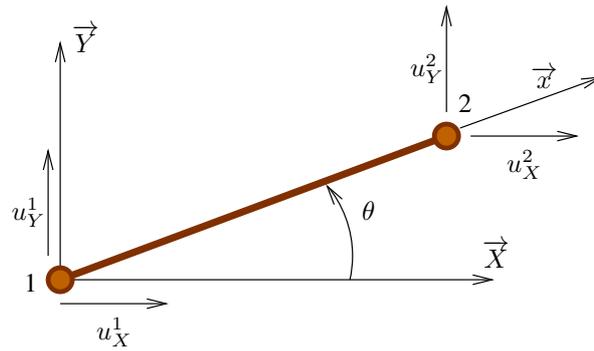


Figure I.7 – Élément barre dans une base globale.

I.6.4 Élément barre pour le calcul des treillis plans

Les barres composant un treillis plan sont positionnées arbitrairement dans l'espace et font des angles différents avec le repère global de la structure (\vec{X}, \vec{Y}) (Fig. I.7).

On note θ l'angle entre l'axe \vec{X} du repère global et l'axe \vec{x} du repère local à la barre. Le vecteur déplacement d'un point de la barre s'écrit dans le repère local

$$\vec{u} = u \vec{x}.$$

Il s'écrit dans le repère global

$$\vec{u} = u^X \vec{X} + u^Y \vec{Y}.$$

En projetant les deux équations précédentes sur \vec{x} il vient

$$u = u^X \cos \theta + u^Y \sin \theta.$$

En notant u_1^X et u_1^Y les déplacements suivant \vec{X} et \vec{Y} du nœud 1 de la barre dans le repère global, et en appliquant la formule précédente au nœud 1, on a

$$u_1 = u_1^X \cos \theta + u_1^Y \sin \theta.$$

En utilisant les mêmes notations pour le nœud 2, on a

$$u_2 = u_2^X \cos \theta + u_2^Y \sin \theta.$$

Ceci peut s'écrire sous la forme matricielle suivante

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ 0 & 0 & \cos \theta & \sin \theta \end{bmatrix}}_{=\mathbf{T}} \underbrace{\begin{bmatrix} u_1^X \\ u_1^Y \\ u_2^X \\ u_2^Y \end{bmatrix}}_{=\underline{Q}}$$

soit

$$\underline{q} = \mathbf{T}\underline{Q}$$

où \underline{Q} est le vecteur des inconnus de déplacements aux nœuds de l'élément dans le repère global et \mathbf{T} est la matrice de transformation passant du repère global au repère local.

Il est possible d'écrire les mêmes relations pour les forces extérieures agissant aux nœuds de l'élément :

$$\vec{F}_1 = F_1 \vec{x} = F_1^X \vec{X} + F_1^Y \vec{Y} \quad \text{et} \quad \vec{F}_2 = F_2 \vec{x} = F_2^X \vec{X} + F_2^Y \vec{Y}$$

donc

$$F_1^X = F_1 \vec{x} \cdot \vec{X} = F_1 \cos \theta \quad \text{et} \quad F_1^Y = F_1 \vec{x} \cdot \vec{Y} = F_1 \sin \theta$$

de même,

$$F_2^X = F_2 \cos \theta \quad \text{et} \quad F_2^Y = F_2 \sin \theta$$

ce qui s'écrit sous forme matricielle

$$\begin{bmatrix} F_1^X \\ F_1^Y \\ F_2^X \\ F_2^Y \end{bmatrix} = \underbrace{\begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & \cos \theta \\ 0 & \sin \theta \end{bmatrix}}_{=\mathbf{T}^T} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$$

La matrice qui apparait pour les forces est la transposée de celle présente dans les relations des déplacements. Finalement, l'équilibre de la barre écrit en fonction des déplacements et des forces dans le repère local à la barre

$$\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \frac{ES}{l} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

devient en fonction des déplacements et des forces dans le repère global

$$\begin{bmatrix} F_1^X \\ F_1^Y \\ F_2^X \\ F_2^Y \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & \cos \theta \\ 0 & \sin \theta \end{bmatrix} \frac{ES}{l} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ 0 & 0 & \cos \theta & \sin \theta \end{bmatrix} \begin{bmatrix} u_1^X \\ u_1^Y \\ u_2^X \\ u_2^Y \end{bmatrix}$$

Tous calculs faits, l'équilibre de la barre en deux dimensions s'écrit

$$\frac{ES}{l} \underbrace{\begin{bmatrix} \cos^2 \theta & \cos \theta \sin \theta & -\cos^2 \theta & -\cos \theta \sin \theta \\ \cos \theta \sin \theta & \sin^2 \theta & -\cos \theta \sin \theta & -\sin^2 \theta \\ -\cos^2 \theta & -\cos \theta \sin \theta & \cos^2 \theta & \cos \theta \sin \theta \\ -\cos \theta \sin \theta & -\sin^2 \theta & \cos \theta \sin \theta & \sin^2 \theta \end{bmatrix}}_{=\mathbf{k}_g} \begin{bmatrix} u_1^X \\ u_1^Y \\ u_2^X \\ u_2^Y \end{bmatrix} = \begin{bmatrix} F_1^X \\ F_1^Y \\ F_2^X \\ F_2^Y \end{bmatrix}$$

où la matrice \mathbf{k}_g est la matrice de rigidité de l'élément barre en deux dimensions.

L'effort normal est déterminé par

$$N = \frac{ES}{l}(u_2 - u_1) = \frac{ES}{l} \left((u_2^X - u_1^X) \cos \theta + (u_2^Y - u_1^Y) \sin \theta \right)$$

II — Problème de référence

II.1 Rappel de mécanique des milieux continus

Soit un corps solide déformable Ω ayant pour surface extérieure $\partial\Omega$ (Fig. II.1). Ce solide subit une force volumique \vec{f} dans Ω et une force surfacique donnée \vec{F}^d sur la surface $\partial_F\Omega$.

Le déplacement \vec{u}^d est imposé sur la surface $\partial_u\Omega$, complémentaire à $\partial_F\Omega$. Les deux surfaces $\partial_u\Omega$ et $\partial_F\Omega$ vérifient :

$$\partial\Omega = \partial_u\Omega \cup \partial_F\Omega \text{ et } \partial_u\Omega \cap \partial_F\Omega = \emptyset.$$

Le problème de mécanique est de déterminer les deux champs inconnus de déplacements \vec{u} et de contraintes σ symétrique ($\sigma = \sigma^T$) sous l'effet des forces appliquées \vec{F}^d et \vec{f} qui vérifient :

1. Les conditions cinématiques :

$$\vec{u} = \vec{u}^d, \quad \forall M \in \partial_u\Omega$$

2. Les équations d'équilibre :

$$\text{div } \sigma + \vec{f} = \vec{0}, \quad \forall M \in \Omega$$

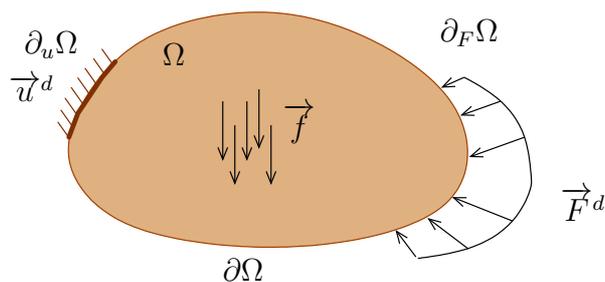


Figure II.1 – Problème de référence en statique.

et

$$\boldsymbol{\sigma} \vec{n} = \vec{F}^d, \quad \forall M \in \partial_F \Omega$$

avec \vec{n} qui est le vecteur normal extérieur à $\partial_F \Omega$.

3. La relation de comportement :

$$\boldsymbol{\sigma} = \mathbb{C} \boldsymbol{\epsilon}(\vec{u})$$

où \mathbb{C} est l'opérateur d'élasticité et $\boldsymbol{\epsilon}$ est l'opérateur linéaire des déformations du champ de déplacement \vec{u} défini par

$$\boldsymbol{\epsilon}(\vec{u}) = \frac{1}{2} \left(\mathbf{Grad}(\vec{u}) + \mathbf{Grad}^T(\vec{u}) \right).$$

Pour un matériau isotrope, les coefficients d'élasticité se réduisent à deux constantes indépendantes (λ, μ) ou (E, ν) où λ et μ sont Les coefficients de Lamé, E est le module d'Young et ν est le coefficient de Poisson. La relation de comportement devient la loi de Hooke, soit

$$\boldsymbol{\sigma} = 2\mu \boldsymbol{\epsilon} + \lambda \text{Tr}(\boldsymbol{\epsilon}) \mathbf{1}$$

où $\mathbf{1}$ désigne la matrice identité. Les coefficients de Lamé (λ, μ) sont reliés à (E, ν) par

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad \text{et} \quad \mu = \frac{E}{2(1+\nu)}.$$

La relation de comportement s'écrit aussi

$$\boldsymbol{\epsilon} = \frac{1+\nu}{E} \boldsymbol{\sigma} - \frac{\nu}{E} \text{Tr}(\boldsymbol{\sigma}) \mathbf{1}.$$

Preuve Équations d'équilibre

Un volume quelconque ω inclu dans Ω subit :

- les efforts surfaciques du vecteur contrainte $\boldsymbol{\sigma} \vec{n}$ sur sa surface extérieure $\partial\omega$,
- les efforts volumiques \vec{f} dans son volume ω

L'équilibre de ce volume se traduit par :

$$\int_{\partial\omega} \boldsymbol{\sigma} \vec{n} \, dS + \int_{\omega} \vec{f} \, d\omega = 0$$

soit en utilisant le théorème de la divergence :

$$\int_{\omega} \vec{\text{div}} \boldsymbol{\sigma} \, d\omega + \int_{\omega} \vec{f} \, d\omega = 0$$

ou encore

$$\int_{\omega} (\vec{\text{div}} \boldsymbol{\sigma} + \vec{f}) \, d\omega = 0$$

Comme cela est valable pour une partie quelconque ω de Ω , on a

$$\forall M \in \Omega, \quad \vec{\text{div}} \boldsymbol{\sigma} + \vec{f} = 0$$

L'équilibre en moment de ω permet de montrer que l'opérateur des contraintes est symétrique. •

II.2 Champ de déplacement cinématiquement admissible

II.2.1 Espace vectoriel des champs de déplacements

On rappelle que l'ensemble \mathcal{V} des champs de déplacement "suffisamment réguliers" définis de Ω vers \mathbb{R}^3 est :

$$\mathcal{V} = \left\{ \begin{array}{l} \vec{v} \text{ "régulier"} : \Omega \rightarrow \mathbb{R}^3 \\ M \mapsto \vec{v}(M) \end{array} \right\}$$

Par soucis de simplicité, la loi interne est notée $+$ dans la suite, la loi externe est notée sans symbole ; enfin l'élément neutre est noté $\vec{0}$.

On peut utiliser une base de \mathbb{R}^3 , par exemple la base canonique $(\vec{e}_1, \vec{e}_2, \vec{e}_3)$ pour représenter un élément \vec{v} de \mathcal{V} :

$$\vec{v}(M) = v_1(M)\vec{e}_1 + v_2(M)\vec{e}_2 + v_3(M)\vec{e}_3$$

On note \underline{v} la représentation matricielle de \vec{v} dans cette base comme

$$\underline{v}(M) = \begin{bmatrix} v_1(M) \\ v_2(M) \\ v_3(M) \end{bmatrix}_{(\vec{e}_1, \vec{e}_2, \vec{e}_3)}$$

où la base est précisée en indice. On utilise la notation indicielle v_i pour manipuler les composantes de \underline{v} . Avec les conventions classiques de notations indicielles, on peut écrire

$$\vec{v}(M) = \sum_{i=1}^3 v_i(M)\vec{e}_i$$

ou encore

$$\vec{v}(M) = v_i(M)\vec{e}_i$$

avec la convention de sommation implicite.

II.2.2 Espace des champs de déplacements cinématiquement admissibles

On définit l'ensemble \mathcal{U} des champs de déplacements cinématiquement admissibles comme :

$$\mathcal{U} = \left\{ \vec{u} \in \mathcal{V} / \vec{u}(M) = \vec{u}^d \quad \forall M \in \partial_u \Omega \right\}$$

Cet espace n'est pas un espace vectoriel car il ne contient pas l'élément neutre, sauf dans le cas particulier où $\vec{u}^d = \vec{0}$.

On dit que \vec{u} est cinématiquement admissible si et seulement si $\vec{u} \in \mathcal{U}$. □

II.3 Champ de contrainte statiquement admissible

II.3.1 Espace vectoriel des champs de contraintes

On pose l'ensemble \mathcal{C} des champs de contraintes "suffisamment réguliers" définis de Ω vers $M_{3,3}^{sym}(\mathbb{R})$ (ensemble des matrices réelles 3×3 symétriques) :

$$\mathcal{C} = \left\{ \begin{array}{l} \sigma \text{ "régulier"} : \Omega \rightarrow M_{3,3}(\mathbb{R}) \\ M \mapsto \sigma(M) \end{array} \right\}$$

Cet espace \mathcal{C} est un espace vectoriel sur \mathbb{R} de dimension infinie.

Exercice II.1 Montrer que \mathcal{C} est un espace vectoriel sur \mathbb{R} . ■

Par soucis de simplicité, la loi interne est notée $+$ dans la suite, la loi externe est notée sans symbole.

On peut utiliser une base de $M_{3,3}^{sym.}(\mathbb{R})$, par exemple la base canonique e_{ij} , pour représenter un élément σ de \mathcal{C} . La représentation matricielle de σ dans la base canonique de \mathbb{R}^3 est

$$\sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}_{(\vec{e}_1, \vec{e}_2, \vec{e}_3)}$$

qui s'écrit dans la base canonique de $M_{3,3}^{sym.}(\mathbb{R})$:

$$\begin{aligned} \sigma = & \underbrace{\sigma_{11} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{=e_{11}} + \underbrace{\sigma_{22} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{=e_{22}} + \underbrace{\sigma_{33} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{=e_{33}} + \underbrace{\sigma_{23} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{=e_{23}} + \underbrace{\sigma_{32} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{=e_{32}} \\ & + \underbrace{\sigma_{13} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{=e_{13}} + \underbrace{\sigma_{31} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}}_{=e_{31}} + \underbrace{\sigma_{12} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{=e_{12}} + \underbrace{\sigma_{21} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{=e_{21}} \end{aligned}$$

Soit aussi en notations indicielles :

$$\sigma = \sigma_{ij} e_{ij}$$

L'opérateur des contraintes étant symétrique, on peut écrire σ à partir de seulement 6 composantes :

$$\sigma = \begin{bmatrix} \sigma_1 & \sigma_6 & \sigma_5 \\ \sigma_6 & \sigma_2 & \sigma_4 \\ \sigma_5 & \sigma_4 & \sigma_3 \end{bmatrix}_{(\vec{e}_1, \vec{e}_2, \vec{e}_3)}$$

ou encore en introduisant la base e'_i :

$$\begin{aligned} \sigma = & \underbrace{\sigma_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{=e'_1} + \underbrace{\sigma_2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{=e'_2} + \underbrace{\sigma_3 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{=e'_3} \\ & + \underbrace{\sigma_4 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{=e'_4} + \underbrace{\sigma_5 \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}}_{=e'_5} + \underbrace{\sigma_6 \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{=e'_6} \end{aligned}$$

En utilisant les notations indicielles, on a avec ces notations :

$$\sigma = \sigma_i e'_i$$

Finalement, on peut noter σ comme

$$\sigma = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix}_{(e'_i)}$$

où la base est précisée en indice. Cette notation est appelée la notation de Voigt. Dans la suite, un élément σ de \mathcal{C} est noté $\underline{\sigma}$ lorsque l'on utilise sa représentation dans la base (e'_i) . Autrement dit, lorsque l'on utilise la notation de Voigt ; on ne précise alors plus la base, elle est implicite :

$$\underline{\sigma} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix} = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix}$$

Exercice II.2 Ecrire l'opérateur des contraintes en notations de Voigt dans le cas des contraintes planes.

Solution :

Dans le cas des contraintes planes dans le plan (\vec{e}_1, \vec{e}_2) , on a

$$\sigma_{33} = \sigma_{13} = \sigma_{23} = 0$$

donc,

$$\sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & 0 \\ \sigma_{12} & \sigma_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix}_{(\vec{e}_1, \vec{e}_2, \vec{e}_3)}$$

soit aussi

$$\sigma = \sigma_{11} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \sigma_{22} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \sigma_{12} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

En écriture de Voigt, l'opérateur des contraintes devient

$$\underline{\sigma} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix}$$

Exercice II.3 Ecrire l'opérateur des contraintes en notations de Voigt dans le cas d'un problème axisymétrique. ■

II.3.2 Espace des champs de contraintes statiquement admissibles

On définit l'ensemble \mathcal{S} des champs de contraintes statiquement admissibles comme :

$$\mathcal{S} = \left\{ \sigma \in \mathcal{C} / \overrightarrow{\text{div}} \sigma + \overrightarrow{f} = 0, \forall M \in \Omega \text{ et } \sigma \overrightarrow{n} = \overrightarrow{F}^d, \forall M \in \partial_F \Omega \right\}$$

ou encore en notations indicielles

$$\mathcal{S} = \left\{ \sigma \in \mathcal{C} / \sigma_{ij,j} + f_i = 0, \forall M \in \Omega \text{ et } \sigma_{ij} n_j = F_i^d, \forall M \in \partial_F \Omega \right\}$$

On dit que σ est statiquement admissible si et seulement si $\sigma \in \mathcal{S}$.

II.4 Écriture du problème de référence

Le couple (\vec{u}, σ) solution du problème est tel que

- \vec{u} est cinématiquement admissible : $\vec{u} \in \mathcal{U}$
- σ est statiquement admissible : $\sigma \in \mathcal{S}$
- \vec{u} et σ vérifient la relation de comportement

$$\sigma = \mathbb{C}\epsilon(\vec{u})$$

avec

$$\epsilon(\vec{u}) = \frac{1}{2}(\mathbf{Grad}(\vec{u}) + \mathbf{Grad}^T(\vec{u}))$$

La solution du problème écrit en déplacement devient :

Le champ de déplacement $\vec{u} \in \mathcal{U}$ est solution du problème si et seulement si $\mathbb{C}\epsilon(\vec{u}) \in \mathcal{S}$

Exercice II.4 Etablir l'ensemble des champs de déplacements cinématiquement admissibles, ainsi que l'ensemble des champs d'efforts normaux statiquement admissibles d'une barre en traction ($x \in [0, L]$), encastrée en $x = 0$, dont l'équation d'équilibre est

$$N'(x) + p = 0 \text{ avec } N(L) = F$$

où N est l'effort normal, p est une charge linéique et F est une force appliquée en $x = L$. On rappelle que le déplacement $u(x)$ est relié à l'effort normal par la relation

$$u'(x) = \frac{N(x)}{ES}$$

où E est le module d'Young et S est l'aire de la section.

En déduire l'écriture de la solution du problème. ■

III — Écriture variationnelle

III.1 Écriture variationnelle de l'ensemble des champs de contraintes admissibles

On souhaite écrire l'ensemble \mathcal{S} sous forme variationnelle. L'ensemble \mathcal{S} est définie par

$$\mathcal{S} = \left\{ \boldsymbol{\sigma} \in \mathcal{C} / \sigma_{ij,j} + f_i = 0 \quad \forall M \in \Omega \quad \text{et} \quad \sigma_{ij} n_j = F_i^d \quad \forall M \in \partial_F \Omega \right\}$$

On introduit l'ensemble \mathcal{U}_0 des champs de déplacements cinématiquement admissibles à zéro tel que

$$\mathcal{U}_0 = \left\{ \vec{u} \in \mathcal{V} / \vec{u}(M) = \vec{0} \quad \forall M \in \partial_u \Omega \right\}$$

Prenons un champ de déplacement $\vec{\delta u} \in \mathcal{U}_0$ quelconque. Ce champ de déplacement est appelé champ de déplacement virtuel, il représente une petite variation de \vec{u} solution du problème (Fig. III.1). On a alors $\boldsymbol{\sigma} \in \mathcal{S}$ équivalent à

$$\sigma_{ij,j} \delta u_i + f_i \delta u_i = 0 \quad \forall M \in \Omega \quad \text{et} \quad \sigma_{ij} n_j = F_i^d \quad \forall M \in \partial_F \Omega$$

Soit

$$\int_{\Omega} \sigma_{ij,j} \delta u_i \, d\Omega + \int_{\Omega} f_i \delta u_i \, d\Omega = 0 \quad \text{et} \quad \sigma_{ij} n_j = F_i^d \quad \forall M \in \partial_F \Omega$$

En intégrant par partie, on a

$$\int_{\Omega} (\sigma_{ij} \delta u_i)_{,j} \, d\Omega - \int_{\Omega} \sigma_{ij} \delta u_{i,j} \, d\Omega + \int_{\Omega} f_i \delta u_i \, d\Omega = 0 \quad \text{et} \quad \sigma_{ij} n_j = F_i^d \quad \forall M \in \partial_F \Omega$$

puis, en utilisant le théorème de Gauss

$$+ \int_{\partial\Omega} \sigma_{ij} n_j \delta u_i \, dS - \int_{\Omega} \sigma_{ij} \delta u_{i,j} \, d\Omega + \int_{\Omega} f_i \delta u_i \, d\Omega = 0 \quad \text{et} \quad \sigma_{ij} n_j = F_i^d \quad \forall M \in \partial_F \Omega$$

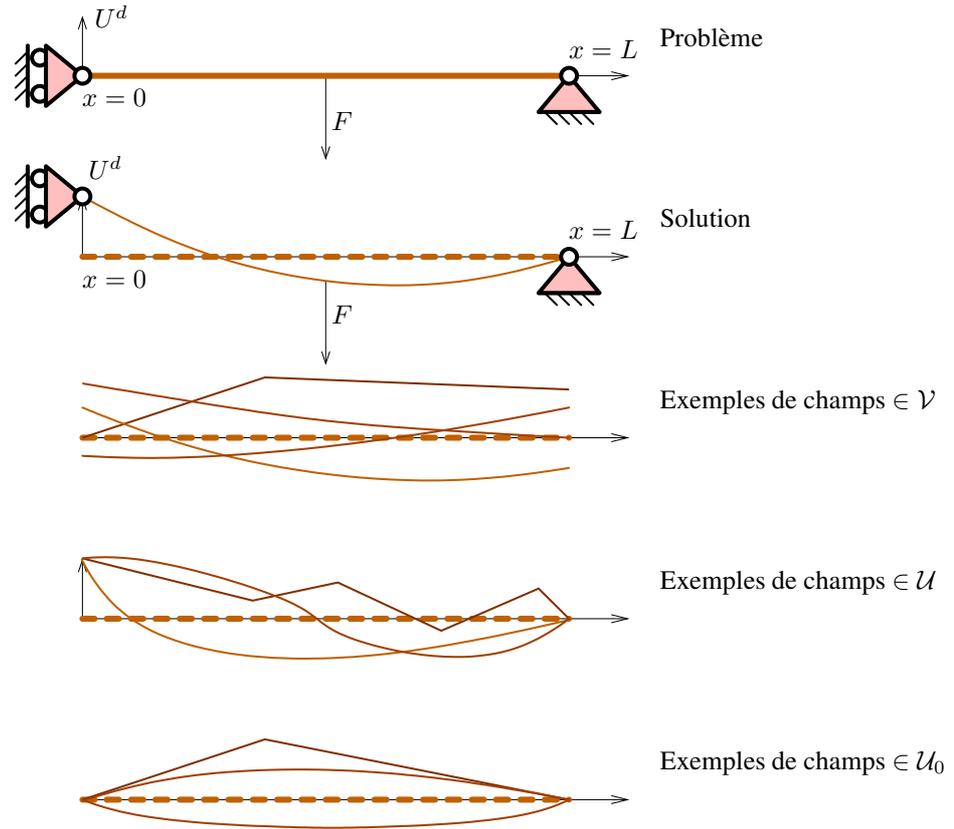


Figure III.1 – Exemples de champs de déplacements de \mathcal{V} , \mathcal{U} et \mathcal{U}_0

En remarquant que $\partial\Omega = \partial_F\Omega \cup \partial_u\Omega$, puis en remplaçant la deuxième équation dans la première sur le bord $\partial_F\Omega$, et enfin en utilisant le fait que $\delta u_i = 0$ sur le bord $\partial_u\Omega$, on a :

$$-\int_{\Omega} \sigma_{ij} \delta u_{i,j} \, d\Omega + \int_{\partial_F\Omega} F_i^d \delta u_i \, dS + \int_{\Omega} f_i \delta u_i \, d\Omega = 0 .$$

En remarquant que $\boldsymbol{\sigma}$ est symétrique, on a la propriété suivante :

$$\sigma_{ij} \delta u_{i,j} = \sigma_{ij} \epsilon_{ij}(\vec{\delta u}) .$$

Preuve

$$\sigma_{ij} \delta u_{i,j} = \frac{1}{2} (\sigma_{ij} \delta u_{i,j} + \sigma_{ij} \delta u_{i,j}) = \frac{1}{2} (\sigma_{ij} \delta u_{i,j} + \sigma_{ji} \delta u_{i,j})$$

car $\boldsymbol{\sigma}$ est symétrique ($\sigma_{ij} = \sigma_{ji}$),

$$\sigma_{ij} \delta u_{i,j} = \frac{1}{2} (\sigma_{ij} \delta u_{i,j} + \sigma_{ij} \delta u_{j,i})$$

car on peut échanger i et j dans le deuxième produit (indices muets),

$$\sigma_{ij} \delta u_{i,j} = \sigma_{ij} \frac{1}{2} (\delta u_{i,j} + \delta u_{j,i}) = \sigma_{ij} \epsilon_{ij}(\vec{\delta u})$$

Finalement, $\boldsymbol{\sigma} \in \mathcal{S}$ est équivalent à

$$\forall \vec{\delta u} \in \mathcal{U}_0, \int_{\Omega} \sigma_{ij} \epsilon_{ij}(\vec{\delta u}) \, d\Omega - \int_{\partial_F\Omega} F_i^d \delta u_i \, dS - \int_{\Omega} f_i \delta u_i \, d\Omega = 0$$

L'ensemble \mathcal{S} s'écrit sous forme variationnelle :

$$\mathcal{S} = \left\{ \boldsymbol{\sigma} \in \mathcal{C} \mid \forall \vec{\delta u} \in \mathcal{U}_0, \int_{\Omega} \sigma_{ij} \epsilon_{ij}(\vec{\delta u}) \, d\Omega - \int_{\partial_F\Omega} F_i^d \delta u_i \, dS - \int_{\Omega} f_i \delta u_i \, d\Omega = 0 \right\}$$

On peut utiliser les notations de Voigt pour écrire plus simplement la formulation variationnelle proposée. La notation de Voigt de σ est donnée par

$$\underline{\sigma} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix} = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix}$$

La notation de Voigt $\underline{\epsilon}$ de ϵ est un peu différente car l'on souhaite que la somme des produits terme à terme de $\underline{\sigma}$ par $\underline{\epsilon}$ donne exactement $\sigma_{ij} \epsilon_{ij}$. Autrement dit, on souhaite que $\underline{\sigma}^T \underline{\epsilon} = \sigma_{ij} \epsilon_{ij}$ en utilisant les notations habituelles de multiplication et de transposition des matrices. En détaillant, on a

$$\sigma_{ij} \epsilon_{ij} = \sigma_{11} \epsilon_{11} + \sigma_{22} \epsilon_{22} + \sigma_{33} \epsilon_{33} + 2\sigma_{23} \epsilon_{23} + 2\sigma_{13} \epsilon_{13} + 2\sigma_{12} \epsilon_{12}$$

$$\sigma_{ij} \epsilon_{ij} = \begin{bmatrix} \sigma_{11} & \sigma_{22} & \sigma_{33} & \sigma_{23} & \sigma_{13} & \sigma_{12} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \epsilon_{23} \\ \epsilon_{13} \\ \epsilon_{12} \end{bmatrix}$$

La matrice qui intervient ci-dessus représente l'opérateur "produit doublement contracté" entre σ et ϵ . Afin d'éviter de faire intervenir cette matrice dans la suite, on note l'opérateur des déformations en notations de Voigt sous la forme :

$$\underline{\epsilon} = \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ 2\epsilon_{23} \\ 2\epsilon_{13} \\ 2\epsilon_{12} \end{bmatrix} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix}$$

où γ_1, γ_2 et γ_3 sont les distorsions angulaires.

On remarque que cette notation est équivalente à changer la base de représentation pour les déformations :

$$\epsilon = \epsilon_{11} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \epsilon_{22} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \epsilon_{33} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} + 2\epsilon_{23} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 \end{bmatrix}$$

$$+ 2\epsilon_{13} \begin{bmatrix} 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{bmatrix} + 2\epsilon_{12} \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

On a alors :

$$\sigma_{ij} \epsilon_{ij} = \underline{\sigma}^T \underline{\epsilon} = \underline{\epsilon}^T \underline{\sigma}$$

L'ensemble \mathcal{S} écrit avec la notation de Voigt est

$$\mathcal{S} = \left\{ \boldsymbol{\sigma} \in \mathcal{C} / \forall \delta \vec{u} \in \mathcal{U}_0, \int_{\Omega} \underline{\epsilon}(\delta \vec{u})^T \underline{\sigma} \, d\Omega - \int_{\partial_F \Omega} \underline{\delta u}^T \underline{F}^d \, dS - \int_{\Omega} \underline{\delta u}^T \underline{f} \, d\Omega = 0 \right\}$$

III.2 Formulation variationnelle en déplacement

III.2.1 Loi de comportement en notations de Voigt

À partir de la loi de Hooke

$$\boldsymbol{\sigma} = \mathbb{C} \boldsymbol{\epsilon} = 2\mu \boldsymbol{\epsilon} + \lambda \text{Tr}(\boldsymbol{\epsilon}) \mathbf{1}$$

on peut écrire l'opérateur \mathbb{C} en notations de Voigt \mathbf{C} tel que

$$\underline{\sigma} = \mathbf{C} \underline{\epsilon}.$$

On a

$$\underline{\sigma} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix} = \underbrace{\begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix}}_{=\mathbf{C}} \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ 2\epsilon_{23} \\ 2\epsilon_{13} \\ 2\epsilon_{12} \end{bmatrix}$$

Exercice III.1 Montrer que dans le cas des contraintes planes, la loi de comportement s'écrit en notations de Voigt sous la forme suivante :

$$\underline{\sigma} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ 2\epsilon_{12} \end{bmatrix} = \mathbf{C} \underline{\epsilon}$$

III.2.2 Opérateur gradient symétrisé en notations de Voigt

On note \mathbf{D} l'opérateur différentiel qui relie le déplacement aux déformations tel que $\underline{\epsilon} = \mathbf{D} \underline{u}$. Les déformations d'un champ de déplacement $\vec{u} \in \mathcal{V}$ s'écrivent

$$\underline{\epsilon}(\vec{u}) = \frac{1}{2} (\mathbf{Grad}(\vec{u}) + \mathbf{Grad}^T(\vec{u})).$$

Cet opérateur a pour expression dans un repère cartésien

$$\underline{\epsilon}(\vec{u}) = \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ 2\epsilon_{23} \\ 2\epsilon_{13} \\ 2\epsilon_{12} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{bmatrix}}_{=\mathbf{D}} \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \mathbf{D} \underline{u}.$$

Exercice III.2 Montrer que l'opérateur différentiel D reliant les déformations au déplacement dans le cas d'un problème plan vaut :

$$\underline{\epsilon} = \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ 2\epsilon_{12} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = D\underline{u}$$

Exercice III.3 Dans le cas d'un problème écrit en repère cylindrique, le déplacement \vec{u} est noté

$$\vec{u} = u\vec{e}_r + v\vec{e}_\theta + w\vec{z}$$

On rappelle que le gradient du déplacement vaut

$$\mathbf{Grad}(\vec{u}) = \begin{bmatrix} \frac{\partial u}{\partial r} & \frac{1}{r} \left(\frac{\partial u}{\partial \theta} - v \right) & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial r} & \frac{1}{r} \left(\frac{\partial v}{\partial \theta} + u \right) & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial r} & \frac{1}{r} \left(\frac{\partial w}{\partial \theta} \right) & \frac{\partial w}{\partial z} \end{bmatrix}_{(\vec{e}_r, \vec{e}_\theta, \vec{k})}$$

Montrer que l'opérateur gradient symétrisé vaut :

$$\underline{\epsilon} = \begin{bmatrix} \epsilon_{rr} \\ \epsilon_{\theta\theta} \\ \epsilon_{zz} \\ 2\epsilon_{rz} \\ 2\epsilon_{z\theta} \\ 2\epsilon_{r\theta} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial r} & 0 & 0 \\ \frac{1}{r} & \frac{1}{r} \frac{\partial}{\partial \theta} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial r} \\ 0 & \frac{\partial}{\partial z} & \frac{1}{r} \frac{\partial}{\partial \theta} \\ \frac{1}{r} \frac{\partial}{\partial \theta} & -\frac{1}{r} + \frac{\partial}{\partial r} & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = D\underline{u}$$

Dans le cas d'un problème axisymétrique, les quantités ne dépendent pas de θ et v est nul. Montrer que l'écriture de D se simplifie et devient

$$\underline{\epsilon} = \begin{bmatrix} \epsilon_{rr} \\ \epsilon_{\theta\theta} \\ \epsilon_{zz} \\ 2\epsilon_{rz} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial r} & 0 \\ \frac{1}{r} & 0 \\ 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial z} & \frac{\partial}{\partial r} \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix} = D\underline{u}.$$

III.2.3 Formulation variationnelle en déplacement, écriture en notations de Voigt

Finalement, en utilisant la relation de comportement, nous venons d'établir que :

Le champ de déplacement $\vec{u} \in \mathcal{U}$ solution du problème vérifie :

$\forall \delta \vec{u} \in \mathcal{U}_0$

$$\int_{\Omega} \underline{\epsilon}^T(\delta \underline{u}) \mathbf{C} \underline{\epsilon}(\underline{u}) \, d\Omega - \int_{\partial_F \Omega} \delta \underline{u}^T \underline{F}^d \, dS - \int_{\Omega} \delta \underline{u}^T \underline{f} \, d\Omega = 0$$

ou bien

$$\int_{\Omega} (\mathbf{D} \delta \underline{u})^T \mathbf{C} \mathbf{D} \underline{u} \, d\Omega - \int_{\partial_F \Omega} \delta \underline{u}^T \underline{F}^d \, dS - \int_{\Omega} \delta \underline{u}^T \underline{f} \, d\Omega = 0$$

Exercice III.4 Établir la formulation variationnelle en déplacement d'une barre en traction ($x \in [0, L]$) dont l'équation d'équilibre est

$$N'(x) + p = 0 \text{ avec } N(L) = F$$

où N est l'effort normal, p est une charge linéique et F est une force appliquée en $x = L$. La barre est encadrée en $x = 0$. On rappelle que le déplacement $u(x)$ est relié à l'effort normal par la relation

$$u'(x) = \frac{N(x)}{ES}$$

où E est le module d'Young et S est l'aire de la section. ■

III.3 Équivalence énergétique

On définit les quantités suivantes :

— Le travail des efforts extérieurs \mathcal{W} dans un champ de déplacement $\vec{v} \in \mathcal{V}$:

$$\mathcal{W}(\vec{v}) = \int_{\Omega} v_i f_i \, d\Omega + \int_{\partial_F \Omega} v_i F_i^d \, dS + \int_{\partial_u \Omega} v_i R_i \, dS$$

où R_i sont les réactions des liaisons sur le bord $\partial_u \Omega$. Pour un champ de déplacement cinématiquement admissible $\vec{u} \in \mathcal{U}$ on a :

$$\mathcal{W}(\vec{u}) = \int_{\Omega} u_i f_i \, d\Omega + \int_{\partial_F \Omega} u_i F_i^d \, dS + \int_{\partial_u \Omega} u_i^d R_i \, dS$$

Pour un champ de déplacement cinématiquement admissible à zéro $\vec{\delta u} \in \mathcal{U}_0$ on a :

$$\mathcal{W}(\vec{\delta u}) = \int_{\Omega} \delta u_i f_i \, d\Omega + \int_{\partial_F \Omega} \delta u_i F_i^d \, dS$$

— L'énergie de déformation \mathcal{E}_D d'un champ de déplacement $\vec{u} \in \mathcal{V}$:

$$\mathcal{E}_D(\vec{u}) = \frac{1}{2} \int_{\Omega} C_{ijkl} \epsilon_{kl}(\vec{u}) \epsilon_{ij}(\vec{u}) \, d\Omega = \frac{1}{2} \int_{\Omega} \underline{\epsilon}^T(\vec{u}) \underline{C} \underline{\epsilon}(\vec{u}) \, d\Omega$$

— L'énergie potentielle \mathcal{E}_p d'un champ de déplacement \vec{u} :

$$\mathcal{E}_p(\vec{u}) = \mathcal{E}_D(\vec{u}) - \mathcal{W}(\vec{u})$$

Théorème III.1 Théorème de la variation virtuelle d'énergie potentielle

$\vec{u} \in \mathcal{U}$ est solution du problème

\Leftrightarrow

\vec{u} minimise l'énergie potentielle \mathcal{E}_p

\Leftrightarrow

Chercher $\vec{u} \in \mathcal{U}$ tel que $\forall \delta \vec{u}, \delta \mathcal{E}_p = 0$

\Leftrightarrow

$\forall \vec{\delta u} \in \mathcal{U}_0$

$$\int_{\Omega} \underline{\epsilon}^T(\delta \underline{u}) \underline{C} \underline{\epsilon}(\underline{u}) \, d\Omega - \int_{\partial_F \Omega} \delta \underline{u}^T \underline{F}^d \, dS - \int_{\Omega} \delta \underline{u}^T \underline{f} \, d\Omega = 0$$

Preuve On suppose que \vec{u} est solution du problème. Lorsque \vec{u} varie de $\vec{\delta u}$ avec $\vec{\delta u} \in \mathcal{U}_0$, l'énergie de déformation varie de $\delta \mathcal{E}_D$:

$$\begin{aligned} \mathcal{E}_D(\vec{u} + \vec{\delta u}) &= \frac{1}{2} \int_{\Omega} C_{ijkl} \epsilon_{ij}(\vec{u} + \vec{\delta u}) \epsilon_{kl}(\vec{u} + \vec{\delta u}) \, d\Omega \\ &= \frac{1}{2} \int_{\Omega} C_{ijkl} \epsilon_{ij}(\vec{u}) \epsilon_{kl}(\vec{u}) \, d\Omega + \frac{1}{2} \int_{\Omega} C_{ijkl} \epsilon_{ij}(\vec{\delta u}) \epsilon_{kl}(\vec{u}) \, d\Omega \\ &\quad + \frac{1}{2} \int_{\Omega} C_{ijkl} \epsilon_{ij}(\vec{u}) \epsilon_{kl}(\vec{\delta u}) \, d\Omega + \underbrace{\frac{1}{2} \int_{\Omega} C_{ijkl} \epsilon_{ij}(\vec{\delta u}) \epsilon_{kl}(\vec{\delta u}) \, d\Omega}_{\text{négligeable}} \end{aligned}$$

Le dernier terme est négligeable devant les trois autres termes. Étant donnée la symétrie de la loi de comportement ($C_{ijkl} = C_{klij}$), le deuxième et le troisième terme sont égaux, l'expression de $\mathcal{E}_D(\vec{u} + \vec{\delta u})$ devient

$$\begin{aligned} \mathcal{E}_D(\vec{u} + \vec{\delta u}) &= \underbrace{\frac{1}{2} \int_{\Omega} C_{ijkl} \epsilon_{ij}(\vec{u}) \epsilon_{kl}(\vec{u}) \, d\Omega}_{=\mathcal{E}_D(\vec{u})} + \underbrace{\int_{\Omega} C_{ijkl} \epsilon_{ij}(\vec{u}) \epsilon_{kl}(\vec{\delta u}) \, d\Omega}_{=\delta \mathcal{E}_D} \\ \mathcal{E}_D(\vec{u} + \vec{\delta u}) &= \mathcal{E}_D(\vec{u}) + \delta \mathcal{E}_D \end{aligned}$$

d'où par identification on a

$$\delta \mathcal{E}_D = \int_{\Omega} C_{ijkl} \epsilon_{ij}(\vec{u}) \epsilon_{kl}(\vec{\delta u}) \, d\Omega$$

Lorsque $\vec{u} \in \mathcal{U}$ varie de $\vec{\delta u}$ avec $\vec{\delta u} \in \mathcal{U}_0$, le travail des forces extérieures varie de $\delta \mathcal{W}$:

$$\begin{aligned} \mathcal{W}(\vec{u} + \vec{\delta u}) &= \underbrace{\int_{\Omega} u_i f_i \, d\Omega + \int_{\partial_F \Omega} u_i F_i^d \, dS + \int_{\partial_u \Omega} u_i^d R_i \, dS}_{=\mathcal{W}(\vec{u})} + \underbrace{\int_{\Omega} \delta u_i f_i \, d\Omega + \int_{\partial_F \Omega} \delta u_i F_i^d \, dS}_{=\delta \mathcal{W}} \\ \mathcal{W}(\vec{u} + \vec{\delta u}) &= \mathcal{W}(\vec{u}) + \delta \mathcal{W} \end{aligned}$$

avec

$$\delta \mathcal{W} = \int_{\partial_F \Omega} \delta u_i F_i^d \, dS + \int_{\Omega} \delta u_i f_i \, d\Omega$$

Or la formulation variationnelle en déplacement implique que

$$\forall \vec{\delta u} \in \mathcal{U}_0, \int_{\Omega} C_{ijkl} \epsilon_{ij}(\vec{u}) \epsilon_{kl}(\vec{\delta u}) \, d\Omega - \int_{\partial_F \Omega} \delta u_i F_i^d \, dS - \int_{\Omega} \delta u_i f_i \, d\Omega = 0$$

c'est à dire

$$\forall \vec{\delta u} \in \mathcal{U}_0, \delta \mathcal{E}_D - \delta \mathcal{W} = 0$$

soit aussi

$$\forall \vec{\delta u} \in \mathcal{U}_0, \delta \mathcal{E}_p = 0$$

III.4 Formes bilinéaire et linéaire, norme en énergie

III.4.1 Forme linéaire

On définit la forme linéaire $L(\vec{v})$ telle que

$$\begin{aligned} L : \mathcal{V} &\rightarrow \mathbb{R} \\ \vec{v} &\mapsto L(\vec{v}) = \int_{\Omega} \vec{v} \cdot \vec{f} \, d\Omega + \int_{\partial_F \Omega} \vec{v} \cdot \vec{F}^d \, dS + \int_{\partial_u \Omega} \vec{v} \cdot \vec{R} \, dS \end{aligned}$$

Le travail des efforts extérieurs \mathcal{W} dans un champ de déplacement $\vec{v} \in \mathcal{V}$ est égal à la forme linéaire :

$$\mathcal{W}(\vec{v}) = L(\vec{v}).$$

III.4.2 Forme bilinéaire

On définit la forme bilinéaire $a(\vec{u}, \vec{v})$ suivante :

$$a : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$$

$$\vec{u}, \vec{v} \mapsto a(\vec{u}, \vec{v}) = \int_{\Omega} \underline{\epsilon}^T(\vec{u}) \mathbf{C} \underline{\epsilon}(\vec{v}) \, d\Omega$$

L'énergie de déformation du champ de déplacement \vec{u} vaut :

$$\mathcal{E}_D(\vec{u}) = \frac{1}{2} a(\vec{u}, \vec{u})$$

On peut montrer que $a(\vec{u}, \vec{v})$ est un produit scalaire, en effet :

- $a(\vec{u}, \vec{v}) = a(\vec{v}, \vec{u})$ car \mathbf{C} est symétrique
- $a(\alpha \vec{u}, \vec{v}) = \alpha a(\vec{u}, \vec{v})$ car $\underline{\epsilon}(\alpha \vec{u}) = \alpha \underline{\epsilon}(\vec{u})$
- $a(\vec{u} + \vec{w}, \vec{v}) = a(\vec{u}, \vec{v}) + a(\vec{w}, \vec{v})$ car $\underline{\epsilon}(\vec{u} + \vec{w}) = \underline{\epsilon}(\vec{u}) + \underline{\epsilon}(\vec{w})$
- $a(\vec{u}, \vec{u}) \geq 0$ car l'énergie de déformation d'un champ de déplacement n'est jamais négative. On dit que $a(\vec{u}, \vec{v})$ est une forme positive.
- $a(\vec{u}, \vec{u}) = 0$ entraîne que \vec{u} est un déplacement de solide rigide (car l'énergie de déformation d'un solide rigide est nulle) ou bien entraîne que \vec{u} est le champ de déplacement nul. Si tous les déplacements rigides sont bloqués alors $a(\vec{u}, \vec{u}) > 0$ si $\vec{u} \neq \vec{0}$, on dit que $a(\vec{u}, \vec{v})$ est une forme définie positive, c'est donc un produit scalaire.

III.4.3 Écriture du problème de référence

On rappelle que le problème de référence écrit en déplacement est :

Trouver $\vec{u} \in \mathcal{U}$ tel que :

$$\forall \delta \vec{u} \in \mathcal{U}_0$$

$$\int_{\Omega} \underline{\epsilon}^T(\delta \vec{u}) \mathbf{C} \underline{\epsilon}(\vec{u}) \, d\Omega - \int_{\partial_F \Omega} \delta \vec{u} \cdot \vec{F}^d \, dS - \int_{\Omega} \delta \vec{u} \cdot \vec{f} \, d\Omega = 0$$

Cela peut s'écrire à l'aide des formes bilinéaire et linéaire introduites comme :

Théorème III.2 $\vec{u} \in \mathcal{V}$ est solution du problème

\Leftrightarrow

$$\vec{u} \in \mathcal{U} \text{ et } \forall \vec{v} \in \mathcal{U}_0, a(\vec{u}, \vec{v}) = L(\vec{v})$$

III.4.4 Norme en énergie

À partir du produit scalaire défini par la forme bilinéaire $a(\vec{u}, \vec{v})$, qui à 2 champs de déplacements de \mathcal{V} associe un scalaire de \mathbb{R} , on construit la norme au sens de l'énergie :

$$\| \vec{u} \|_E = \left(a(\vec{u}, \vec{u}) \right)^{\frac{1}{2}}$$

On remarque que

$$\mathcal{E}_D(\vec{u}) = \frac{1}{2} \int_{\Omega} \underline{\epsilon}^T(\vec{u}) \mathbf{C} \underline{\epsilon}(\vec{u}) \, d\Omega = \frac{1}{2} a(\vec{u}, \vec{u}) = \frac{1}{2} \| \vec{u} \|_E^2$$

IV — Espace d'approximation E.F.

IV.1 Introduction

Le problème de statique linéaire consiste à chercher la solution en déplacement \vec{u} dans un espace de dimension infinie \mathcal{V} . Sauf quelques cas particuliers, la solution exacte, notée \vec{u}^{ex} n'est pas accessible.

L'idée de la méthode des éléments finis est de chercher une solution approchée, notée \vec{u}^h (h fait référence à la taille d'un élément fini) dans un espace de dimension finie n noté \mathcal{V}^h .

L'objet de ce chapitre est de présenter la construction de cet espace vectoriel.

IV.2 Fonction à une variable

On considère l'espace vectoriel sur \mathbb{R} des fonctions à une variable définies de $[A, B]$ vers \mathbb{R} :

$$\mathcal{V} = \left\{ \begin{array}{l} v \text{ "régulier"} : [A, B] \rightarrow \mathbb{R} \\ x \mapsto v(x) \end{array} \right\}$$

Plusieurs fonctions de \mathcal{V} sont représentées sur la figure IV.1 ; il en existe une infinité.

L'objectif est de construire un sous-espace vectoriel \mathcal{V}^h de dimension finie n de \mathcal{V} . Pour construire ce sous-espace, on peut se donner n fonctions simples $\varphi_i(x) \in \mathcal{V}$ avec $i \in [1, 2, \dots, n]$. Ces n fonctions engendrent un sous-espace vectoriel de \mathcal{V} . Si elles sont linéairement indépendantes, elles forment alors une base de \mathcal{V}^h . Elles sont appelées les fonctions de base (ou fonction de forme) de l'espace d'approximation \mathcal{V}^h . Pour les construire simplement, on découpe l'intervalle $[A, B]$ en $n - 1$ intervalles appelés éléments. On introduit les n points $x_i \in [A, B]$ tels que $x_1 = A$ et $x_n = B$, ces points sont appelés les nœuds.

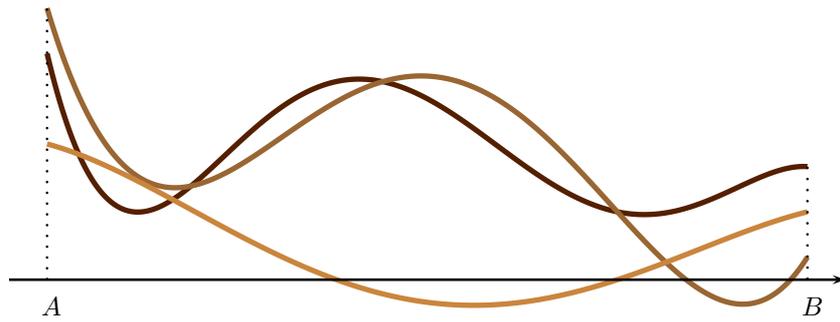


Figure IV.1 – Fonctions de \mathcal{V} à une variable définie sur $[A, B]$.

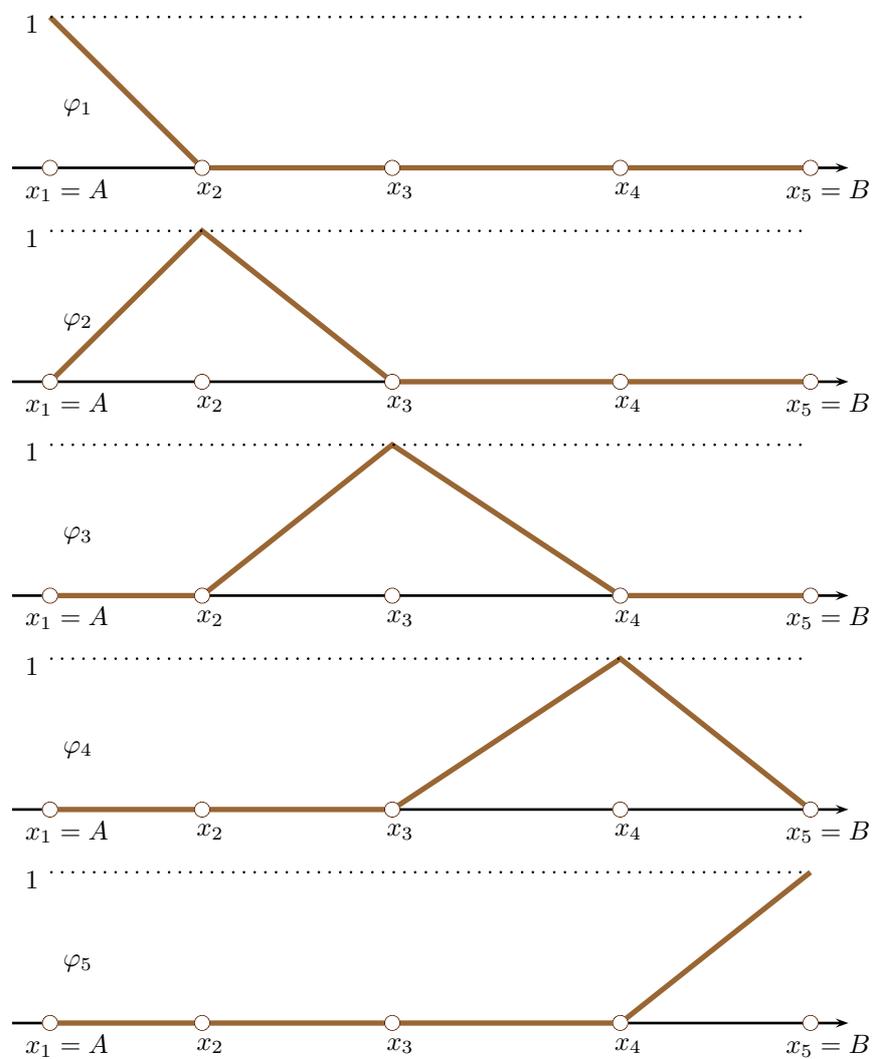


Figure IV.2 – Fonctions de base de l'intervalle $[A, B]$.

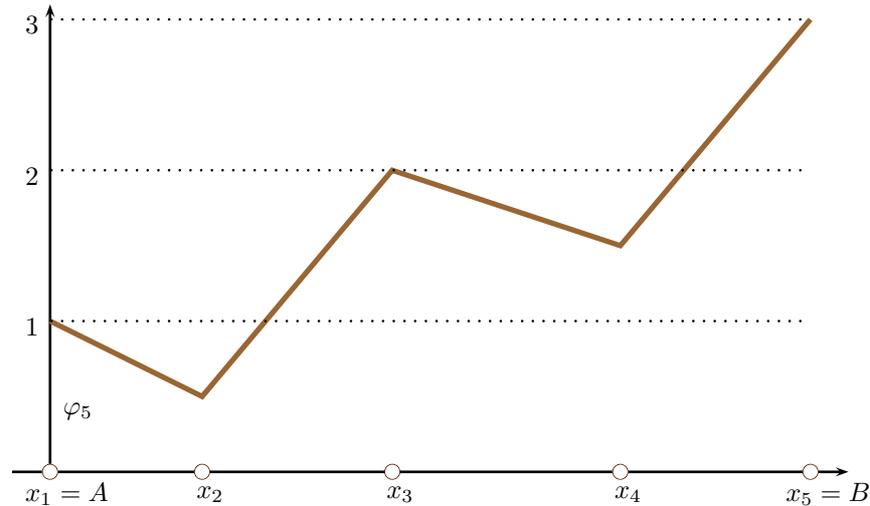


Figure IV.3 – Exemple de combinaison linéaire des fonctions de base pour $u_1 = 1$; $u_2 = 0,5$; $u_3 = 2$; $u_4 = 1,5$; $u_5 = 3$.

La figure IV.2 présente un intervalle $[A, B]$ découpé en 4 éléments par 5 nœuds. La fonction φ_i est associée au nœud i , elle vaut 1 en ce nœud et 0 aux autres. Elle s'annule dans les éléments qui ne sont pas directement adjacents (ou connectés) au nœud i . Sur l'exemple proposé, comme on a choisi 5 nœuds, on construit 5 fonctions φ_i . Ces 5 fonctions engendrent un sous-espace vectoriel \mathcal{V}^h de l'espace vectoriel \mathcal{V} de dimension infinie. On voit clairement que les fonctions sont linéairement indépendantes, c'est à dire que l'on ne peut pas en obtenir une à partir des autres ; ces fonctions forment donc une base de \mathcal{V}^h qui est alors de dimension 5. Dans le cas plus général de n points, le sous-espace vectoriel \mathcal{V}^h peut être écrit comme :

$$\mathcal{V}^h = \text{vect} \{ \varphi_1, \varphi_2, \dots, \varphi_n \}$$

Tout élément (vecteur) de \mathcal{V}^h s'écrit comme une combinaison linéaire des fonctions φ_i :

$$u^h = \alpha_1 \varphi_1 + \alpha_2 \varphi_2 + \dots + \alpha_n \varphi_n$$

En écrivant la valeur de u^h au nœud i , on a (sans sommation implicite)

$$u^h(x_i) = \alpha_1 \underbrace{\varphi_1(x_i)}_{=0} + \alpha_2 \underbrace{\varphi_2(x_i)}_{=0} + \dots + \alpha_i \underbrace{\varphi_i(x_i)}_{=1} + \dots + \alpha_n \underbrace{\varphi_n(x_i)}_{=0} = \alpha_i$$

On remarque que le coefficient multiplicatif α_i de la fonction de base associée au nœud i vaut la valeur de la fonction u^h à ce nœud. En posant $u^h(x_i) = u_i$, on a

$$u^h = u_1 \varphi_1 + u_2 \varphi_2 + \dots + u_n \varphi_n$$

On donne sur la figure IV.3 un exemple de combinaison linéaire des fonctions de base avec 5 points pour les valeurs suivantes :

$$u_1 = 1 \ ; \ u_2 = 0,5 \ ; \ u_3 = 2 \ ; \ u_4 = 1,5 \ ; \ u_5 = 3$$

On peut écrire plus simplement u^h dans la base des φ_i comme :

$$u^h = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}_{(\varphi_i)}$$

Afin de simplifier les écritures, on note \underline{U} la représentation de u^h dans la base des fonctions de forme φ_i ; la base étant implicite :

$$\underline{U} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$$

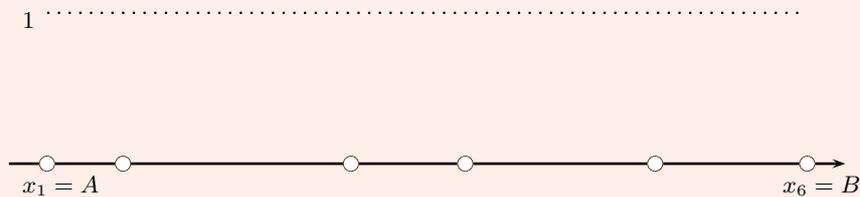
En notant la matrice Φ contenant les fonctions de forme sous la forme :

$$\Phi = [\varphi_1 \ \varphi_2 \ \cdots \ \varphi_n]$$

on peut alors écrire

$$u^h = \Phi \underline{U}.$$

Exercice IV.1 Dessiner les 6 fonctions de base φ_i construites sur le maillage suivant. Tracer la fonction u^h associée à $u_1 = 1,5$; $u_2 = 0,5$; $u_3 = -1$; $u_4 = 1$; $u_5 = 2$; $u_6 = 0$.



IV.3 Fonction à deux variables

On considère l'ensemble des fonctions à deux variables définies d'un domaine Ω du plan vers \mathbb{R} :

$$\mathcal{V} = \left\{ \begin{array}{l} v \text{ "régulier"} : \Omega \rightarrow \mathbb{R} \\ (x, y) \mapsto v(x, y) \end{array} \right\}$$

Une fonction de \mathcal{V} est représentée sur la figure IV.4.

L'objectif est de construire un sous-espace vectoriel \mathcal{V}^h de dimension finie n de \mathcal{V} . Pour construire ce sous-espace, on peut se donner n fonctions simples $\varphi_i(x, y) \in \mathcal{V}$ avec $i \in [1, 2, \dots, n]$. Ces n fonctions engendrent un sous-espace vectoriel de \mathcal{V} . Si elles sont linéairement indépendantes, elles forment alors une base de \mathcal{V}^h . Elles sont appelées les fonctions de base (ou fonction de forme) de l'espace d'approximation \mathcal{V}^h .

Pour les construire simplement, on découpe le domaine (pièce ou structure) en éléments. Ici, on choisit par exemple un découpage à l'aide de triangles (Fig. IV.5). Ce pavage est appelé maillage, les triangles en sont les éléments, les sommets des triangles les nœuds. Chaque nœud est en contact

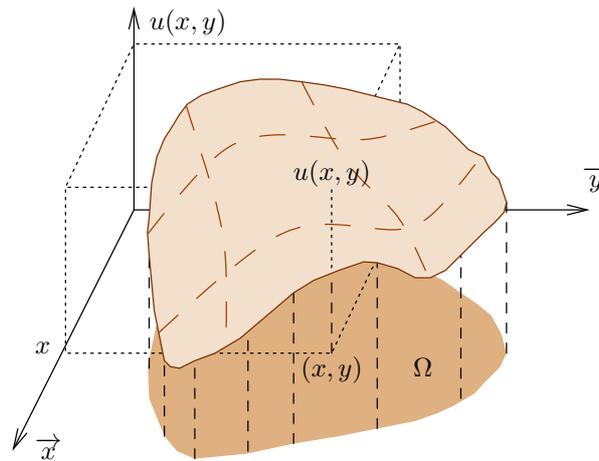


Figure IV.4 – Une fonction à deux variables définie sur un domaine Ω du plan.

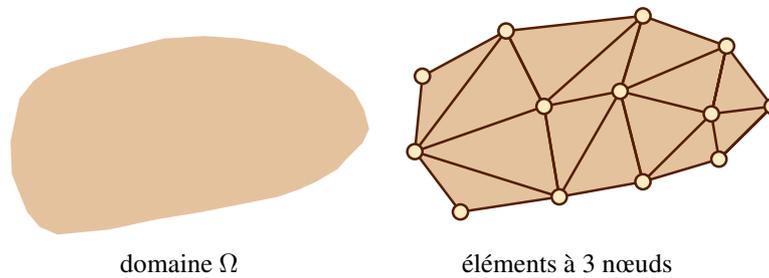


Figure IV.5 – Pavage d'un domaine Ω du plan avec des triangles à 3 nœuds.

avec plusieurs éléments et chaque élément est construit à partir de plusieurs nœuds (3 pour des triangles à 3 nœuds).

On associe à chaque nœud du maillage une fonction de base $\varphi_i(x, y)$. Le nombre de nœuds n du maillage donne alors la dimension de \mathcal{V}^h . La fonction de base associée à un nœud s'étend sur tous les éléments en contact avec ce nœud, elle est définie par morceaux. Les règles de construction des fonctions de base φ_i sont :

- φ_i vaut 1 au nœud i : $\varphi_i(x_i, y_i)=1$;
- φ_i vaut 0 pour tous les autres nœuds $j \neq i$: $\varphi_i(x_j, y_j)=0$;
- φ_i vaut 0 dans tous les éléments non connectés directement au nœud i ;
- φ_i est continue et dérivable.

Sur l'exemple de la figure IV.6, la fonction de forme du nœud 1 est représentée. Elle vérifie toutes les règles précédentes.

La fonction de base $\varphi_i(x, y)$ est donc construite par morceaux. Pour l'exemple des triangles à 3 nœuds utilisé ici, chaque portion de $\varphi_i(x, y)$ est un plan passant par 3 points.

L'espace \mathcal{V}^h peut alors être définie comme l'espace engendré par les fonctions φ_i :

$$\mathcal{V}^h = \text{vect} \{ \varphi_1, \varphi_2, \dots, \varphi_n \}$$

Autrement dit, tout élément $u^h \in \mathcal{V}^h$ s'écrit sous la forme d'une combinaison linéaire des φ_i :

$$u^h = \alpha_1 \varphi_1 + \alpha_2 \varphi_2 + \dots + \alpha_n \varphi_n$$

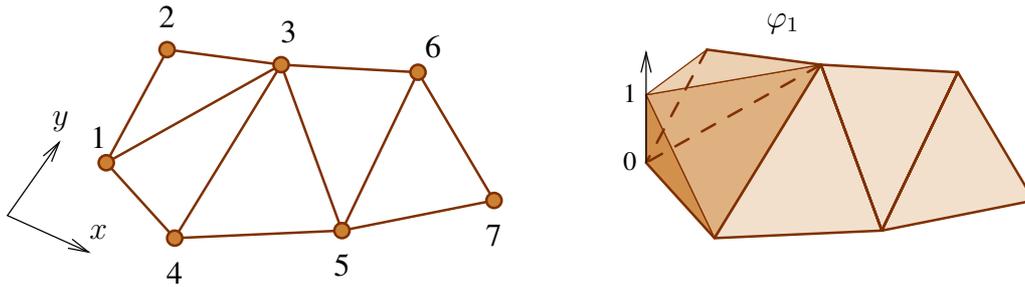


Figure IV.6 – Fonction de forme associée à un nœud.

En écrivant la valeur de u^h au nœud i , on a

$$u^h(x_i, y_i) = \alpha_1 \underbrace{\varphi_1(x_i, y_i)}_{=0} + \alpha_2 \underbrace{\varphi_2(x_i, y_i)}_{=0} + \dots + \alpha_i \underbrace{\varphi_i(x_i, y_i)}_{=1} + \dots + \alpha_n \underbrace{\varphi_n(x_i, y_i)}_{=0} = \alpha_i$$

On remarque que le coefficient multiplicatif α_i de la fonction de base associée au nœud i vaut la valeur de la fonction u^h à ce nœud. En posant $u^h(x_i, y_i) = u_i$, on a

$$u^h = u_1\varphi_1 + u_2\varphi_2 + \dots + u_n\varphi_n$$

On peut écrire plus simplement u^h dans la base des φ_i comme :

$$u^h = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}_{(\varphi_i)}$$

Afin de simplifier les écritures, on note \underline{U} la représentation de u^h dans la base des fonctions de forme φ_i ; la base étant implicite :

$$\underline{U} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$$

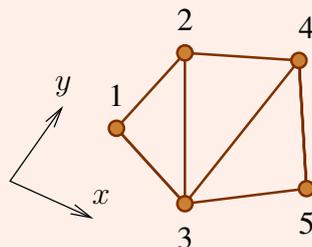
En notant la matrice Φ contenant les fonctions de forme sous la forme :

$$\Phi = [\varphi_1 \ \varphi_2 \ \dots \ \varphi_n]$$

on peut alors écrire

$$u^h = \Phi \underline{U}.$$

Exercice IV.2 Pour le pavage suivant constitué de triangles, tracer les 5 fonctions de base $\varphi_i(x, y)$. Tracer la fonction u^h associée à $u_1 = 0,5$; $u_2 =$; $u_3 = 0,5$; $u_4 = 2$; $u_5 = 3$.



IV.4 Fonction vectorielle à deux variables

On considère l'ensemble des fonctions vectorielles à deux variables définies d'un domaine Ω du plan vers \mathbb{R}^2 :

$$\mathcal{V} = \left\{ \begin{array}{l} \vec{v} \text{ "régulier"} : \Omega \rightarrow \mathbb{R}^2 \\ (x, y) \mapsto \vec{v}(x, y) \end{array} \right\}$$

On construit dans un premier temps les fonctions $\varphi_i(x, y)$ de la même façon que ce qui est fait pour le cas d'une fonction à deux variables. On construit ensuite l'ensemble \mathcal{V}^h à partir des couples de fonctions vectorielles $\varphi_i(x, y)\vec{e}_1, \varphi_i(x, y)\vec{e}_2 \in \mathcal{V}$ où i varie de 1 à n . L'espace vectoriel \mathcal{V}^h est alors engendré par ces fonctions :

$$\mathcal{V}^h = \text{vect} \{ \varphi_1(x, y)\vec{e}_1, \varphi_1(x, y)\vec{e}_2, \dots, \varphi_n(x, y)\vec{e}_1, \varphi_n(x, y)\vec{e}_2 \}$$

La dimension de \mathcal{V}^h est $2n$.

Tout champ de déplacement de \mathcal{V}^h s'écrit sous la forme :

$$\vec{u}^h(x, y) = u_1\varphi_1(x, y)\vec{e}_1 + v_1\varphi_1(x, y)\vec{e}_2 + \dots + u_n\varphi_n(x, y)\vec{e}_1 + v_n\varphi_n(x, y)\vec{e}_2$$

ou encore

$$\underline{u}^h = \Phi \underline{U}$$

avec

$$\Phi = \begin{bmatrix} \varphi_1 & 0 & \varphi_2 & 0 & \dots & \varphi_n & 0 \\ 0 & \varphi_1 & 0 & \varphi_2 & \dots & 0 & \varphi_n \end{bmatrix}$$

où Φ est appelée la matrice des fonctions de forme et \underline{U} est la représentation de $\underline{u}^h(x, y)$ dans la base des fonctions de forme :

$$\underline{U} = \begin{bmatrix} u_1 \\ v_1 \\ \vdots \\ u_n \\ v_n \end{bmatrix}$$

IV.5 Fonction à trois variables

On considère l'ensemble des fonctions à trois variables définies d'un domaine Ω de l'espace vers \mathbb{R} :

$$\mathcal{V} = \left\{ \begin{array}{l} v \text{ "régulier"} : \Omega \rightarrow \mathbb{R} \\ (x, y, z) \mapsto v(x, y, z) \end{array} \right\}$$

Pour construire le sous-espace \mathcal{V}^h , on se donne n fonctions simples $\varphi_i(x, y, z) \in \mathcal{V}$ avec $i \in [1, 2, \dots, n]$. Ces n fonctions engendrent \mathcal{V}^h un sous-espace vectoriel de \mathcal{V} . Elles forment une base de \mathcal{V}^h si elle sont linéairement indépendantes (famille libre).

Pour les construire simplement, on découpe le domaine tri-dimensionnel en éléments. Ici, on choisit par exemple un découpage à l'aide de tétraèdres (Fig. IV.7). Chaque élément est construit à partir de plusieurs nœuds, 4 pour des tétraèdres à 4 nœuds.

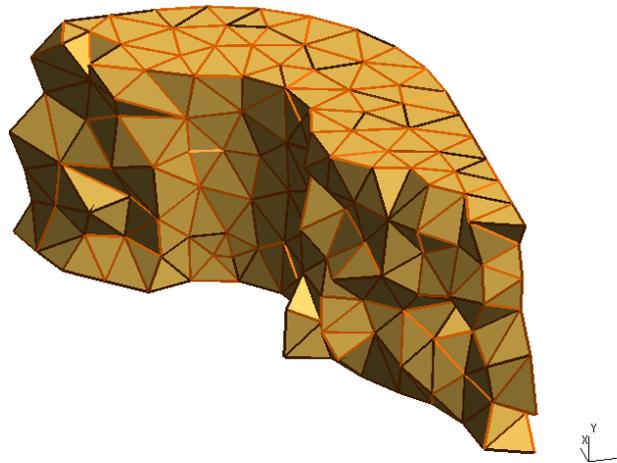


Figure IV.7 – Maillage d'un domaine tridimensionnel avec des tétraèdres à 4 nœuds.

La fonction $\varphi_i(x, y, z)$ est construite par morceaux de façon à valoir 1 au nœud i , zéro aux autres nœuds et zéro dans tous les éléments non connectés au nœud i . Le détail de ces fonctions est donné dans la suite du cours.

L'espace \mathcal{V}^h est alors définie comme l'espace engendré par les fonctions $\varphi_i(x, y, z)$:

$$\mathcal{V}^h = \text{vect} \{ \varphi_1(x, y, z), \varphi_2(x, y, z), \dots, \varphi_n(x, y, z) \}$$

La dimension de \mathcal{V}^h est n . Autrement dit, tout élément $u^h \in \mathcal{V}^h$ s'écrit sous la forme d'une combinaison linéaire des $\varphi_i(x, y, z)$:

$$u^h(x, y, z) = u_1 \varphi_1(x, y, z) + u_2 \varphi_2(x, y, z) + \dots + u_n \varphi_n(x, y, z)$$

où $u^h(x_i, y_i, z_i) = u_i$

On peut écrire plus simplement u^h dans la base des φ_i comme :

$$u^h = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}_{(\varphi_i)}$$

Afin de simplifier les écritures, on note \underline{U} la représentation de u^h dans la base des fonctions de forme φ_i ; la base étant implicite :

$$\underline{U} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$$

IV.6 Fonction vectorielle à trois variables

On considère l'ensemble des fonctions vectorielles à trois variables définies d'un domaine Ω de l'espace vers \mathbb{R}^3 :

$$\mathcal{V} = \left\{ \begin{array}{l} \vec{v} \text{ "régulier"} : \Omega \rightarrow \mathbb{R}^3 \\ (x, y, z) \mapsto \vec{v}(x, y, z) \end{array} \right\}$$

On construit dans un premier temps les fonctions $\varphi_i(x, y, z)$ de la même façon que ce qui est fait pour le cas d'une fonction à trois variables. On construit ensuite l'ensemble \mathcal{V}^h à partir des triplets de fonctions vectorielles $\varphi_i(x, y, z)\vec{e}_1, \varphi_i(x, y, z)\vec{e}_2, \varphi_i(x, y, z)\vec{e}_3 \in \mathcal{V}$ où i varie de 1 à n . L'espace vectoriel \mathcal{V}^h est alors engendré par ces fonctions :

$$\mathcal{V}^h = \text{vect} \{ \varphi_1(x, y, z)\vec{e}_1, \varphi_1(x, y, z)\vec{e}_2, \varphi_1(x, y, z)\vec{e}_3, \dots, \\ \varphi_n(x, y, z)\vec{e}_1, \varphi_n(x, y, z)\vec{e}_2, \varphi_n(x, y, z)\vec{e}_3 \}$$

La dimension de \mathcal{V}^h est $3n$.

Tout champ de déplacement de \mathcal{V}^h s'écrit sous la forme :

$$\vec{u}^h(x, y, z) = u_1\varphi_1(x, y, z)\vec{e}_1 + v_1\varphi_1(x, y, z)\vec{e}_2 + w_1\varphi_1(x, y, z)\vec{e}_3 + \dots \\ + u_n\varphi_n(x, y, z)\vec{e}_1 + v_n\varphi_n(x, y, z)\vec{e}_2 + w_n\varphi_n(x, y, z)\vec{e}_3$$

ou encore

$$\underline{u}^h = \Phi \underline{U}$$

avec

$$\Phi = \begin{bmatrix} \varphi_1 & 0 & 0 & \dots & \varphi_n & 0 & 0 \\ 0 & \varphi_1 & 0 & \dots & 0 & \varphi_n & 0 \\ 0 & 0 & \varphi_1 & \dots & 0 & 0 & \varphi_n \end{bmatrix}$$

où Φ est appelée la matrice des fonctions de forme et \underline{U} est la représentation de \underline{u}^h dans la base des fonctions de forme :

$$\underline{U} = \begin{bmatrix} u_1 \\ v_1 \\ w_1 \\ \vdots \\ u_n \\ v_n \\ w_n \end{bmatrix}$$

V — Construction de la base E.F.

V.1 Construction des fonctions de forme dans un élément

V.1.1 Introduction

Les fonctions de forme φ_i d'un maillage sont définies par morceaux sur plusieurs éléments. Pour tous les points M appartenant à un élément Ω_e (e est le numéro de l'élément), les fonctions de \mathcal{V}^h ne s'écrivent qu'à partir des fonctions de forme associées aux nœuds connectés à l'élément Ω_e . On note \mathcal{V}_e l'espace vectoriel des fonctions régulières définies du domaine élémentaire Ω_e vers \mathbb{R} .

$$\mathcal{V}_e = \left\{ \begin{array}{l} v \text{ "régulier"} : \Omega_e \rightarrow \mathbb{R} \\ M \mapsto v(M) \end{array} \right\}$$

Les fonctions de forme des nœuds connectés à l'élément e forment un sous-espace vectoriel de \mathcal{V}_e de dimension finie égale au nombre de nœuds n connectés à l'élément e . On peut définir cet espace \mathcal{V}_e^h comme étant engendré par les n fonctions φ_i .

V.1.2 Fonctions de forme dans un élément unidimensionnel

Pour le cas d'une fonction à une variable définie sur l'intervalle $[A, B]$, on peut écrire tout élément de \mathcal{V}^h sous la forme

$$u^h(x) = u_1\varphi_1(x) + u_2\varphi_2(x) + \dots + u_n\varphi_n(x)$$

où n est le nombre de nœuds du maillage. Les fonctions de forme étant définies par morceaux, on a par exemple pour le maillage de la figure V.1 :

$$\forall x \in [x_1, x_2], u^h(x) = u_1\varphi_1(x) + u_2\varphi_2(x)$$

$$\forall x \in [x_2, x_3], u^h(x) = u_2\varphi_2(x) + u_3\varphi_3(x)$$

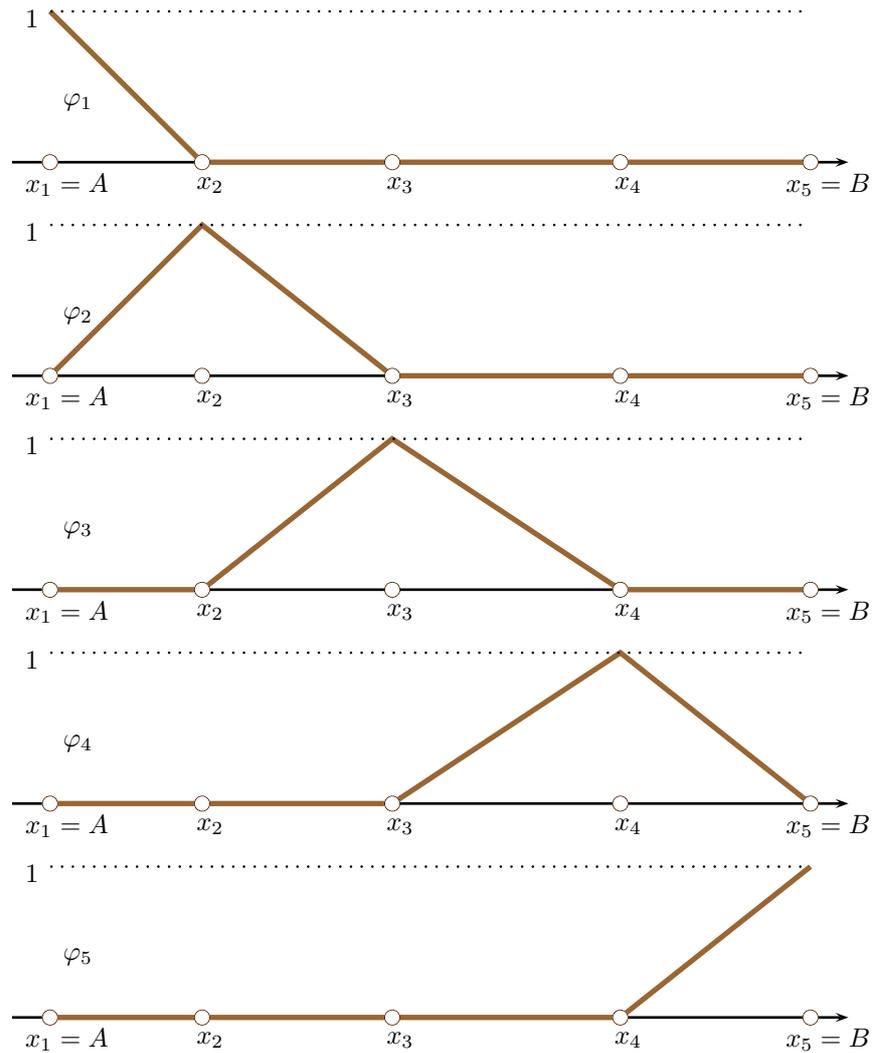


Figure V.1 – Fonctions de base de l'intervalle $[A, B]$.

$$\forall x \in [x_3, x_4], u^h(x) = u_3\varphi_3(x) + u_4\varphi_4(x)$$

$$\forall x \in [x_4, x_5], u^h(x) = u_4\varphi_4(x) + u_5\varphi_5(x)$$

Les fonctions de forme φ_i sont représentées élément par élément sur la figure V.2. On remarque que l'on peut représenter de façon générale les fonctions de forme dans un élément générique allant de a à b comme deux portions de droites comme indiqué sur la figure V.3. Les deux fonctions de base dans cet élément sont notées φ_a et φ_b .

On considère le sous-espace vectoriel \mathcal{V}_e^h de \mathcal{V}^h restreint à l'élément e allant de a à b qui coïncident avec les deux nœuds de l'élément e du maillage initial :

$$\mathcal{V}_e^h = \text{vect} \{ \varphi_a(x), \varphi_b(x) \}$$

La dimension de \mathcal{V}_e^h est 2. Une fonction u_e^h appartenant à \mathcal{V}_e^h s'écrit comme une combinaison linéaire de φ_a et φ_b :

$$\forall x \in [a, b], u_e^h(x) = u_a\varphi_a(x) + u_b\varphi_b(x)$$

On cherche à écrire les fonctions de forme comme des polynômes en x . La base canonique polynomiale contenant 2 termes est $[1 \ x]$: On peut alors ré-écrire \mathcal{V}_e^h comme un espace engendré par

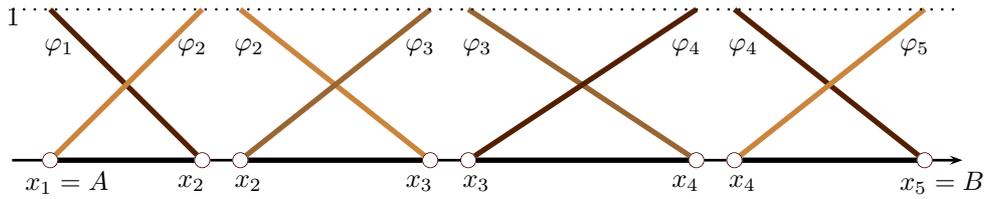


Figure V.2 – Fonctions de base pour chaque élément.

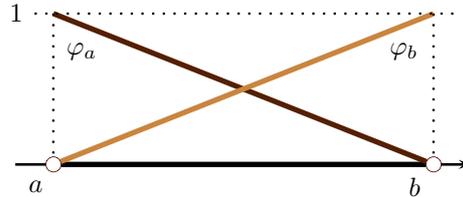


Figure V.3 – Fonctions de base pour un élément $[a, b]$.

la base $[1 \ x]$:

$$\mathcal{V}_e^h = \text{vect} \{1, x\}$$

Finalement, le passage de la base canonique polynômiale $[1 \ x]$ à la base des fonctions de forme $[\varphi_a(x) \ \varphi_b(x)]$ est un changement de base. Il suffit d'écrire la base canonique dans la base des fonctions de forme pour trouver les expressions de φ_a et φ_b :

$$\forall x \in [a, b], 1 = 1\varphi_a(x) + 1\varphi_b(x)$$

$$\forall x \in [a, b], x = a\varphi_a(x) + b\varphi_b(x)$$

soit le système linéaire à deux inconnues suivant à résoudre :

$$\begin{bmatrix} 1 & 1 \\ a & b \end{bmatrix} \begin{bmatrix} \varphi_a(x) \\ \varphi_b(x) \end{bmatrix} = \begin{bmatrix} 1 \\ x \end{bmatrix}$$

ce qui donne le résultat :

$$\varphi_a(x) = \frac{x-b}{a-b} = \frac{b-x}{l} \quad \text{et} \quad \varphi_b(x) = \frac{x-a}{b-a} = \frac{x-a}{l}$$

avec $l = b - a$. En faisant un changement d'origine sur l'axe x tel que $x' = x - a$, on a :

$$\varphi_a(x') = 1 - \frac{x'}{l} \quad \text{et} \quad \varphi_b(x') = \frac{x'}{l}$$

Ces 2 fonctions sont tracées sur la figure V.4.

V.1.3 Fonctions de forme dans un élément du plan

Pour le cas d'une fonction à deux variables définie de Ω vers \mathbb{R} , on utilise par exemple un pavage ou maillage de Ω en triangles. On considère le sous-espace vectoriel \mathcal{V}_e^h de \mathcal{V}^h restreint à l'élément e (domaine noté Ω_e) défini par les points a, b et c (Fig. V.5) :

$$\mathcal{V}_e^h = \text{vect} \{ \varphi_a(x, y), \varphi_b(x, y), \varphi_c(x, y) \}$$

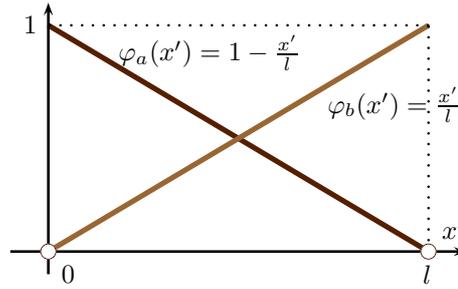


Figure V.4 – Fonctions de forme d'un élément à 2 nœuds.

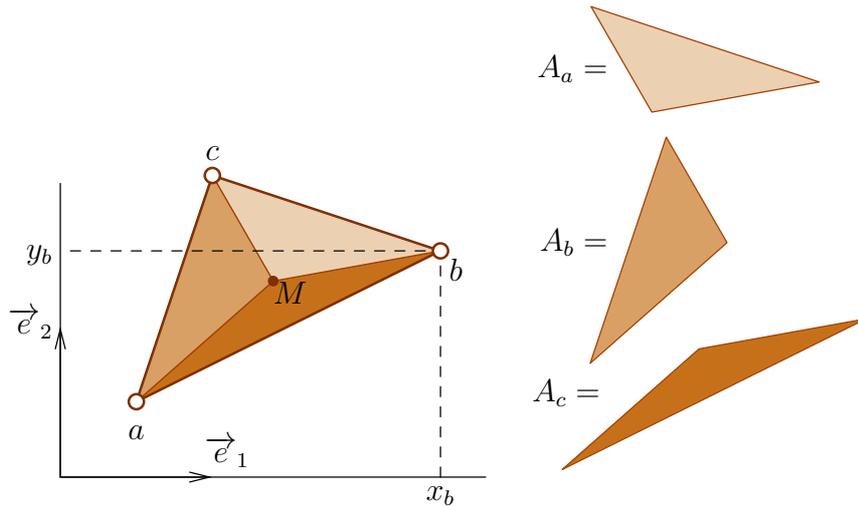


Figure V.5 – Triangle à 3 nœuds.

La dimension de \mathcal{V}_e^h est 3. Une fonction u_e^h appartenant à \mathcal{V}_e^h s'écrit comme une combinaison linéaire de φ_a , φ_b et φ_c :

$$\forall x, y \in \Omega_e, u_e^h(x, y) = u_a \varphi_a(x, y) + u_b \varphi_b(x, y) + u_c \varphi_c(x, y)$$

On cherche à écrire les fonctions de forme comme des polynômes à deux variables x et y . Comme la dimension de \mathcal{V}_e^h est 3, on cherche une base de \mathcal{V}_e^h ayant trois termes.

Le choix de la base $[1 \ x \ y]$ contenant 3 termes conduit à un élément à 3 nœuds, c'est à dire un triangle à 3 nœuds. La base est symétrique, on peut échanger x et y sans changer la base. Des bases plus complètes sont possibles, ces éléments sont détaillés dans la suite.

Finalement, le passage de la base canonique polynômiale $[1 \ x \ y]$ à la base des fonctions de forme $[\varphi_a \ \varphi_b \ \varphi_c]$ est un changement de base. Il suffit d'écrire la base canonique dans la base des fonctions de forme pour trouver les expressions de φ_a , φ_b et φ_c :

$$\forall x, y \in \Omega_e, 1 = 1\varphi_a + 1\varphi_b + 1\varphi_c$$

$$\forall x, y \in \Omega_e, x = x_a\varphi_a + x_b\varphi_b + x_c\varphi_c$$

$$\forall x, y \in \Omega_e, y = y_a\varphi_a + y_b\varphi_b + y_c\varphi_c$$

où (x_a, y_a) , (x_b, y_b) et (x_c, y_c) sont les coordonnées des 3 nœuds a , b et c du triangle.

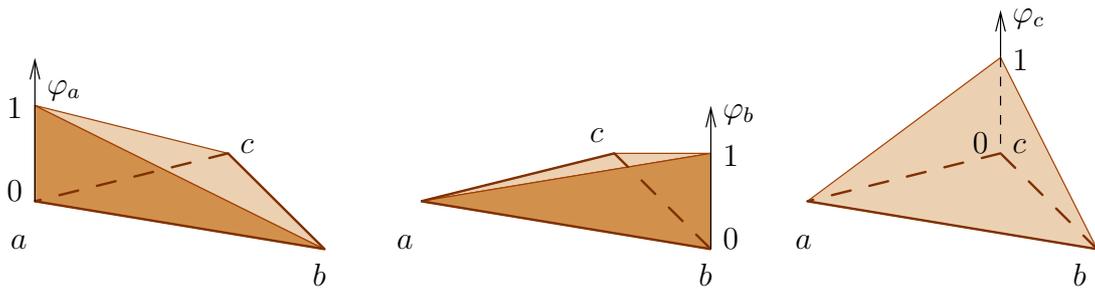


Figure V.6 – Fonctions de forme du triangle à 3 nœuds.

Ce système de trois équations à trois inconnues s’écrit :

$$\begin{bmatrix} 1 & 1 & 1 \\ x_a & x_b & x_c \\ y_a & y_b & y_c \end{bmatrix} \begin{bmatrix} \varphi_a(x, y) \\ \varphi_b(x, y) \\ \varphi_c(x, y) \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ y \end{bmatrix}$$

On peut résoudre ce système par la méthode de Cramer, il faut alors calculer les quantités suivantes :

$$2A = \det \begin{bmatrix} 1 & 1 & 1 \\ x_a & x_b & x_c \\ y_a & y_b & y_c \end{bmatrix} \quad 2A_a = \det \begin{bmatrix} 1 & 1 & 1 \\ x & x_b & x_c \\ y & y_b & y_c \end{bmatrix}$$

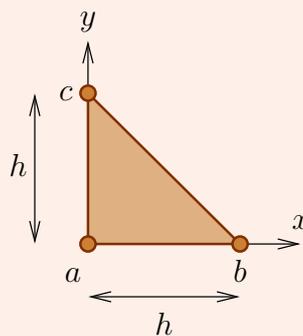
$$2A_b = \det \begin{bmatrix} 1 & 1 & 1 \\ x_a & x & x_c \\ y_a & y & y_c \end{bmatrix} \quad 2A_c = \det \begin{bmatrix} 1 & 1 & 1 \\ x_a & x_b & x \\ y_a & y_b & y \end{bmatrix}$$

où A est l’aire du triangle passant par les 3 nœuds, A_a est l’aire du sous-triangle Mbc , A_b est l’aire du sous-triangle Mac et A_c est l’aire du sous-triangle Mab (Fig. V.5). La solution du système s’écrit :

$$\varphi_a(x, y) = \frac{A_a(x, y)}{A} \quad \varphi_b(x, y) = \frac{A_b(x, y)}{A} \quad \varphi_c(x, y) = \frac{A_c(x, y)}{A}$$

Les fonctions sont tracées sur la figure V.6.

Exercice V.1 Donner les expressions des fonctions de forme associées au triangle à 3 nœuds (a , b et c) suivant. Tracer ces fonctions.



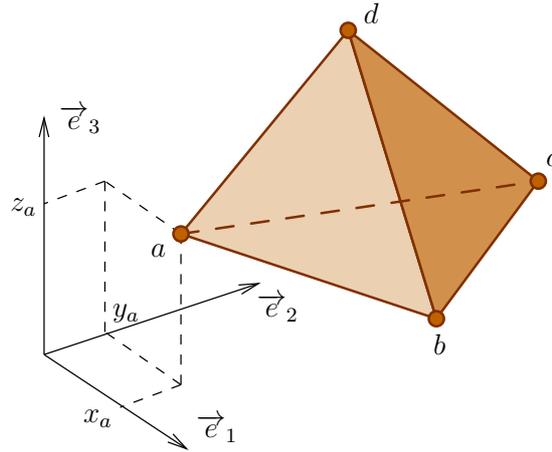


Figure V.7 – Élément tétraèdre à 4 nœuds

V.1.4 Fonctions de forme dans un élément tridimensionnel

Dans le cas d'une fonction à trois variables définie de Ω vers \mathbb{R} , on utilise par exemple un pavage ou maillage de Ω en tétraèdres. On considère le sous-espace vectoriel \mathcal{V}_e^h de \mathcal{V}^h restreint à l'élément e (domaine noté Ω_e) défini par les points a, b, c et d (Fig. V.7) :

$$\mathcal{V}_e^h = \text{vect} \{ \varphi_a(x, y, z), \varphi_b(x, y, z), \varphi_c(x, y, z), \varphi_d(x, y, z) \}$$

La dimension de \mathcal{V}_e^h est 4. Une fonction u_e^h appartenant à \mathcal{V}_e^h s'écrit comme une combinaison linéaire de $\varphi_a, \varphi_b, \varphi_c$ et φ_d :

$$\forall x, y, z \in \Omega_e, u_e^h(x, y, z) = u_a \varphi_a(x, y, z) + u_b \varphi_b(x, y, z) + u_c \varphi_c(x, y, z) + u_d \varphi_d(x, y, z)$$

On cherche à écrire les fonctions de forme comme des polynômes à trois variables x, y et z . Comme la dimension de \mathcal{V}_e^h est 4, on cherche une base de \mathcal{V}_e^h ayant quatre termes. Le choix de la base $[1 \ x \ y \ z]$ contenant 4 termes conduit à un élément à 4 nœuds, c'est à dire un tétraèdre à 4 nœuds. La base est symétrique, on peut échanger x, y et z sans changer la base. Des bases plus complètes sont possibles, ces éléments sont détaillés dans la suite.

En appliquant la même méthode que précédemment pour trouver les fonctions de base dans l'élément, on abouti au système suivant qui permet de trouver les fonctions de forme dans un élément :

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ x_a & x_b & x_c & x_d \\ y_a & y_b & y_c & y_d \\ z_a & z_b & z_c & z_d \end{bmatrix} \begin{bmatrix} \varphi_a(x, y, z) \\ \varphi_b(x, y, z) \\ \varphi_c(x, y, z) \\ \varphi_d(x, y, z) \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix}$$

La solution de ce système est

$$\varphi_i(x, y, z) = \frac{1}{6v} (\alpha_i + \beta_i x + \gamma_i y + \delta_i z)$$

avec les coefficients calculés comme les déterminants suivants :

$$\alpha_i = (-1)^{p+1} \begin{vmatrix} x_j & x_k & x_l \\ y_j & y_k & y_l \\ z_j & z_k & z_l \end{vmatrix} \quad \beta_i = (-1)^p \begin{vmatrix} 1 & 1 & 1 \\ y_j & y_k & y_l \\ z_j & z_k & z_l \end{vmatrix}$$

i	j	k	l	p	i	j	k	l	p	i	j	k	l	p	i	j	k	l	p
a	b	c	d	1	b	a	c	d	0	c	a	b	d	1	d	a	b	c	0

Tableau V.1 – Permutations pour les fonctions de forme de l'élément tétraèdre à 4 nœuds

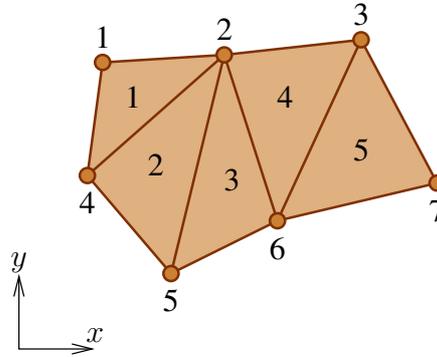


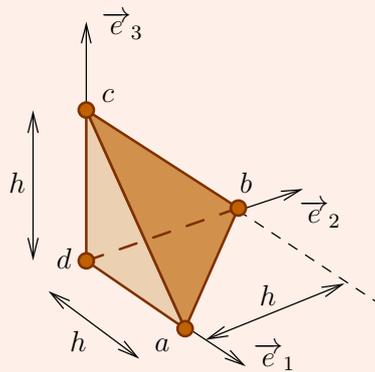
Figure V.8 – Exemple de maillage.

$$\gamma_i = (-1)^{p+1} \begin{vmatrix} 1 & 1 & 1 \\ x_j & x_k & x_l \\ z_j & z_k & z_l \end{vmatrix} \quad \delta_i = (-1)^p \begin{vmatrix} 1 & 1 & 1 \\ x_j & x_k & x_l \\ y_j & y_k & y_l \end{vmatrix}$$

où v est le volume de l'élément et en utilisant la permutation donnée dans le tableau V.1. Le volume v de l'élément est donné par le déterminant suivant :

$$v = \frac{1}{6} \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_a & x_b & x_c & x_d \\ y_a & y_b & y_c & y_d \\ z_a & z_b & z_c & z_d \end{vmatrix}$$

Exercice V.2 Donner les expressions des fonctions de forme associées au tétraèdre à 4 nœuds (a, b, c et d) suivant. A l'aide d'une échelle de couleur, dessiner les 4 fonctions de forme.



V.2 Table de connectivité et coordonnées des nœuds

La définition du maillage nécessite 2 tableaux de valeurs : les coordonnées des nœuds et la connectivité des éléments.

L'exemple de la figure V.8 montre un maillage de triangles à 3 nœuds constitués de 5 éléments et de 7 nœuds. La table de connectivité donne les numéros des nœuds connectés à chaque élément. Le nombre de lignes de ce tableau est le nombre d'éléments du maillage (5 ici), le nombre de colonnes est le nombre de nœuds par élément (3 ici) :

Numéro d'élément	Nœud	Nœud	Nœud
	a	b	c
1	1	2	4
2	4	5	2
3	6	2	5
4	3	2	6
5	3	6	7

Les coordonnées des nœuds sont données dans un tableau dont le nombre de lignes est le nombre de nœuds du maillage (7 ici) et le nombre de colonnes est la dimension de l'espace de modélisation (2 ici, x et y) :

Numéro du Nœud	Coord.	Coord.
	x	y
1	x_1	y_1
2	x_2	y_2
3	x_3	y_3
4	x_4	y_4
5	x_5	y_5
6	x_6	y_6
7	x_7	y_7

Les algorithmes de construction d'un maillage à partir d'une définition géométrique numérique de la pièce (CAO) ne sont pas décrit dans ce cours.

Exercice V.3 Tracer le maillage constitué de triangles associé à ces coordonnées et cette connectivité :

Numéro d'élément	Nœud	Nœud	Nœud	Numéro du Nœud	Coord.	Coord.
	a	b	c		x	y
1	1	4	3	1	0.0	0.0
2	1	2	4	2	1.0	0.0
3	2	5	4	3	0.0	1.0
				4	1.0	1.0
				5	2.0	1.0

VI — Problème discrétisé

VI.1 Définition de l'erreur de discrétisation

On définit l'erreur de discrétisation en déplacement comme la différence entre la solution de référence $\vec{u}^{ex} \in \mathcal{V}$ et la solution éléments finis $\vec{u}^h \in \mathcal{V}^h$:

$$\vec{e} = \vec{u}^{ex} - \vec{u}^h$$

On remarque que $\vec{e} \in \mathcal{V}$.

On définit l'erreur en énergie comme la norme énergétique de \vec{e} :

$$e = \|\vec{e}\|_E$$

Si l'erreur e est nulle alors \vec{e} est le champ de déplacement nul (définition d'une norme) : la solution éléments finis est égale à la solution exacte : $\vec{u}^h = \vec{u}^{ex}$.

VI.2 Écriture du problème discrétisé

VI.2.1 Rappel des différents espaces introduits

Une représentation des différents espaces introduits est donnée sur la figure VI.1. Deux cas peuvent se présenter :

- Cas où $\vec{u}^d \neq \vec{0}$ alors $\mathcal{U} \cap \mathcal{U}_0 = \emptyset$
- Cas où $\vec{u}^d = \vec{0}$, alors $\mathcal{U} = \mathcal{U}_0$

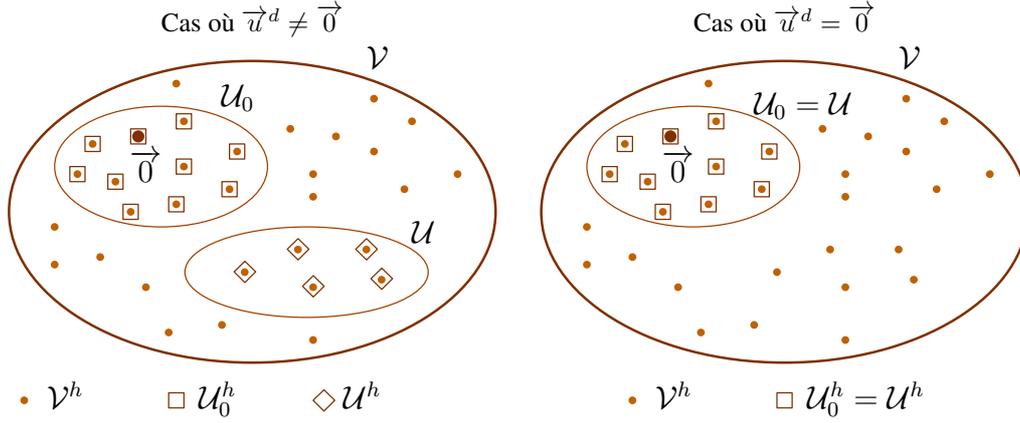


Figure VI.1 – Différents espaces introduits

VI.2.2 Solution en déplacement

La solution éléments finis $\vec{u}^h \in \mathcal{V}^h$ minimise l'erreur avec la solution de référence $\vec{u}^{ex} \in \mathcal{V}$. Autrement dit, l'erreur est la partie de la solution $\vec{u}^{ex} \in \mathcal{V}$ que $\vec{u}^h \in \mathcal{V}^h$ ne peut pas représenter ; cela vient du fait que \mathcal{V}^h est de dimension n et est inclue dans \mathcal{V} .

Cela revient à dire que l'erreur $\vec{e} \in \mathcal{V}$ est orthogonale à tout champ de déplacement $\vec{v}^h \in \mathcal{V}^h$ au sens du produit scalaire $a(\cdot, \cdot)$ (Fig. VI.2) :

$$\forall \vec{v}^h \in \mathcal{V}^h, a(\vec{v}^h, \vec{e}) = 0$$

Comme $\mathcal{U}_0^h \subset \mathcal{V}^h$, a fortiori, on a :

$$\forall \vec{v}^h \in \mathcal{U}_0^h, a(\vec{v}^h, \vec{e}) = 0$$

c'est à dire

$$\forall \vec{v}^h \in \mathcal{U}_0^h, a(\vec{v}^h, \vec{u}^{ex} - \vec{u}^h) = 0 \text{ ou encore } a(\vec{v}^h, \vec{u}^h) = a(\vec{v}^h, \vec{u}^{ex})$$

Or, la solution exacte vérifie :

$$\forall \vec{v} \in \mathcal{U}_0, a(\vec{v}, \vec{u}^{ex}) = L(\vec{v})$$

Comme $\mathcal{U}_0^h \subset \mathcal{U}_0$, on aussi

$$\forall \vec{v}^h \in \mathcal{U}_0^h, a(\vec{v}^h, \vec{u}^{ex}) = L(\vec{v}^h)$$

Finalement, en remplaçant dans l'équation précédente, on a :

$$\forall \vec{v}^h \in \mathcal{U}_0^h, a(\vec{v}^h, \vec{u}^h) = L(\vec{v}^h)$$

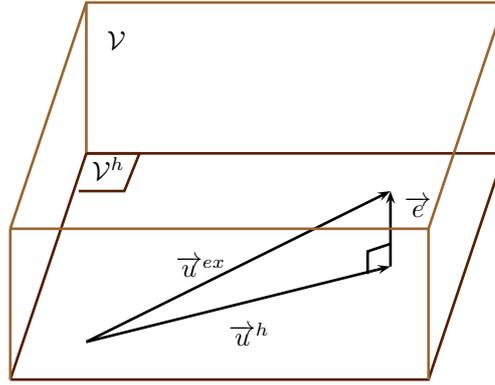
La solution éléments finis $\vec{u}^h \in \mathcal{U}^h$ vérifie

$$\forall \vec{v}^h \in \mathcal{U}_0^h, a(\vec{v}^h, \vec{u}^h) = L(\vec{v}^h)$$

Ce qui s'écrit aussi :

La solution éléments finis $\vec{u}^h \in \mathcal{U}^h$ vérifie

$$\forall \vec{v}^h \in \mathcal{U}_0^h, \int_{\Omega} \underline{\underline{\epsilon}}^T(\vec{v}^h) \mathbf{C} \underline{\underline{\epsilon}}(\vec{u}^h) d\Omega = \int_{\partial_F \Omega} \vec{v}^h \cdot \vec{F}^d dS + \int_{\Omega} \vec{v}^h \cdot \vec{f} d\Omega = 0$$

Figure VI.2 – L'erreur est orthogonale à l'espace \mathcal{V}^h .

VI.2.3 Écriture matricielle du problème discrétisé

On sait que \vec{u}^h (ou \underline{u}^h dans une base donnée) s'écrit à partir des fonctions de base comme :

$$\underline{u}^h = \Phi \underline{U}$$

Le calcul de $\epsilon(\underline{u}^h)$ donne :

$$\epsilon(\underline{u}^h) = \mathbf{D} \Phi \underline{U} = \mathbf{B} \underline{U}$$

où \mathbf{B} est appelé l'opérateur gradient (symétrisé) discrétisé et vaut $\mathbf{B} = \mathbf{D} \Phi$. De même, on a pour l'écriture du champ virtuel \vec{v}^h :

$$\underline{v}^h = \Phi \underline{V} \text{ et } \epsilon(\underline{v}^h) = \mathbf{B} \underline{V}$$

Dans le cas général tridimensionnel, on a :

$$\underline{u}^h(M) = \begin{bmatrix} u^h(M) \\ v^h(M) \\ w^h(M) \end{bmatrix}_{(\vec{e}_1, \vec{e}_2, \vec{e}_3)} = \Phi \underline{U}$$

avec

$$\Phi = \begin{bmatrix} \dots & \varphi_i & 0 & 0 & \dots \\ \dots & 0 & \varphi_i & 0 & \dots \\ \dots & 0 & 0 & \varphi_i & \dots \end{bmatrix} \text{ et } \underline{U} = \begin{bmatrix} \vdots \\ u_i \\ v_i \\ w_i \\ \vdots \end{bmatrix}$$

où i désigne le nœud i . On rappelle que l'opérateur différentiel \mathbf{D} vaut

$$\mathbf{D} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{bmatrix}$$

Donc l'opérateur B devient

$$B = D\Phi = \begin{bmatrix} \dots & \frac{\partial \varphi_i}{\partial x} & 0 & 0 & \dots \\ \dots & 0 & \frac{\partial \varphi_i}{\partial y} & 0 & \dots \\ \dots & 0 & 0 & \frac{\partial \varphi_i}{\partial z} & \dots \\ \dots & 0 & \frac{\partial \varphi_i}{\partial z} & \frac{\partial \varphi_i}{\partial y} & \dots \\ \dots & \frac{\partial \varphi_i}{\partial z} & 0 & \frac{\partial \varphi_i}{\partial x} & \dots \\ \dots & \frac{\partial \varphi_i}{\partial y} & \frac{\partial \varphi_i}{\partial x} & 0 & \dots \end{bmatrix}$$

Le problème discrétisé est alors de trouver $\vec{u}^h \in \mathcal{U}^h$, ou de façon équivalente $\underline{U} \in \mathcal{U}^h$ tel que, $\forall \vec{v}^h \in \mathcal{U}_0^h$, ou de façon équivalente $\forall \underline{V} \in \mathcal{U}_0^h$, on ait

$$\int_{\Omega} \underline{V}^T \mathbf{B}^T \mathbf{C} \mathbf{B} \underline{U} \, d\Omega = \int_{\partial_F \Omega} \underline{V}^T \Phi^T \underline{F}^d \, dS + \int_{\Omega} \underline{V}^T \Phi^T \underline{f} \, d\Omega = 0$$

Dans cette dernière expression, on remarque que \underline{U} et \underline{V} ne dépendent pas du point M dans le domaine Ω , ni sur le bord $\partial_F \Omega$, on peut donc les sortir des intégrales.

On pose

$$\mathbf{K} = \int_{\Omega} \mathbf{B}^T \mathbf{C} \mathbf{B} \, d\Omega$$

et

$$\underline{F} = \int_{\partial_F \Omega} \Phi^T \underline{F}^d \, dS + \int_{\Omega} \Phi^T \underline{f} \, d\Omega$$

La matrice \mathbf{K} est appelée la matrice de rigidité de la structure. Cette matrice représente l'application bilinéaire $a(\cdot, \cdot)$ introduite précédemment pour les champs de déplacements éléments finis appartenant à \mathcal{V}^h , autrement dit :

$$\forall \vec{u}^h \in \mathcal{U}^h, \forall \vec{v}^h \in \mathcal{U}^h, \quad a(\vec{u}^h, \vec{v}^h) = \underline{U}^T \mathbf{K} \underline{V} = \underline{V}^T \mathbf{K} \underline{U}$$

Le vecteur $\underline{F} \in \mathcal{V}^h$ est appelé le vecteur des forces aux nœuds. Ce vecteur est la projection dans \mathcal{V}^h des forces extérieures à la structure, il représente l'application linéaire $L(\cdot)$ pour les champs de déplacements éléments finis appartenant à \mathcal{V}^h , autrement dit :

$$\forall \vec{v}^h \in \mathcal{U}^h, \quad L(\vec{v}^h) = \underline{V}^T \underline{F}$$

On a finalement le problème discrétisé suivant :

Trouver $\underline{U} \in \mathcal{U}^h$ (cinématiquement admissible) tel que $\forall \underline{V} \in \mathcal{U}_0^h$, on ait

$$\underline{V}^T \mathbf{K} \underline{U} = \underline{V}^T \underline{F}$$

VI.3 Résolution du problème discrétisé

VI.3.1 Séparation des équations

On décompose le vecteur des déplacements en deux parties : les déplacements libres $\underline{U}_{[l]}$ (inconnus) et les déplacements prescrits $\underline{U}_{[p]}^d$ (connus). Un champ de déplacement $\vec{u}^h \in \mathcal{U}$ s'écrit alors

$$\underline{U} = \begin{bmatrix} \underline{U}_{[l]} \\ \underline{U}_{[p]}^d \end{bmatrix}$$

Avec ces notations, $[l]$ désigne l'ensemble des degrés de libertés libres, $[p]$ désigne l'ensemble des degrés de libertés prescrits.

Un champ de déplacement virtuel $\vec{v}^h \in \mathcal{U}_0$ s'écrit

$$\underline{V} = \begin{bmatrix} \underline{V}_{[l]} \\ \underline{0}_{[p]} \end{bmatrix}$$

De même, on décompose le vecteur des forces généralisées

$$\underline{F} = \begin{bmatrix} \underline{F}_{[l]}^d \\ \underline{F}_{[p]} \end{bmatrix}$$

où $\underline{F}_{[l]}^d$ correspond aux forces appliquées sur la structure (connues) et $\underline{F}_{[p]}$ correspond aux réactions d'appuis (inconnues).

On décompose la matrice de rigidité de la même façon :

$$\underline{K} = \begin{bmatrix} \underline{K}_{[ll]} & \underline{K}_{[lp]} \\ \underline{K}_{[lp]}^T & \underline{K}_{[pp]} \end{bmatrix}$$

où $\underline{K}_{[ll]}$ correspond aux lignes et colonnes de \underline{K} des déplacements libres, $\underline{K}_{[pp]}$ correspond aux lignes et colonnes de \underline{K} des déplacements bloqués, $\underline{K}_{[lp]}$ correspond aux lignes des déplacements libres et colonnes des déplacements bloqués.

VI.3.2 Résolution en déplacements

Le problème à résoudre est :

Trouver $\underline{U}_{[l]}$ tel que $\forall \underline{V}_{[l]}$, on ait

$$[\underline{V}_{[l]}^T \quad \underline{0}_{[p]}^T] \begin{bmatrix} \underline{K}_{[ll]} & \underline{K}_{[lp]} \\ \underline{K}_{[lp]}^T & \underline{K}_{[pp]} \end{bmatrix} \begin{bmatrix} \underline{U}_{[l]} \\ \underline{U}_{[p]}^d \end{bmatrix} = [\underline{V}_{[l]}^T \quad \underline{0}_{[p]}^T] \begin{bmatrix} \underline{F}_{[l]}^d \\ \underline{F}_{[p]} \end{bmatrix}$$

Ce qui revient à trouver $\underline{U}_{[l]}$ tel que $\forall \underline{V}_{[l]}$, on ait

$$\underline{V}_{[l]}^T (\underline{K}_{[ll]} \underline{U}_{[l]} + \underline{K}_{[lp]} \underline{U}_{[p]}^d) = \underline{V}_{[l]}^T \underline{F}_{[l]}^d$$

Puisque cette équation est valable quelque soit $\underline{V}_{[l]}$, cela revient à résoudre le système suivant :

$$\underline{K}_{[ll]} \underline{U}_{[l]} = \underline{F}_{[l]}^d - \underline{K}_{[lp]} \underline{U}_{[p]}^d$$

La solution de ce système donne les déplacements inconnus $\underline{U}_{[l]}$.

VI.3.3 Réactions aux appuis

Les forces de réactions $\underline{F}_{[p]}$ sont les efforts extérieurs sur les bords $\partial_u \Omega$ où les déplacements sont imposés. Ces efforts assurent l'équilibre global de la structure, ils sont inconnus.

Pour les calculer, on échange virtuellement les frontières $\partial_F \Omega$ et $\partial_u \Omega$. Le problème s'écrit de la même façon que précédemment à la différence que \underline{V} devient :

$$\underline{V} = \begin{bmatrix} \underline{0}_{[l]} \\ \underline{V}_{[p]} \end{bmatrix}$$

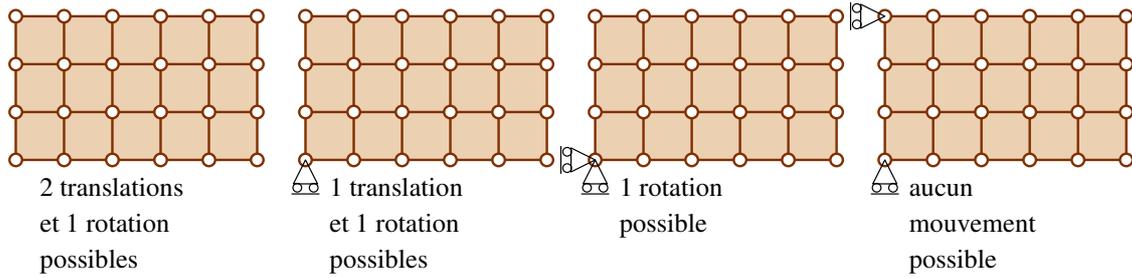


Figure VI.3 – Exemple de blocage des modes de déplacements de solide rigide en bidimensionnel.

Dans ce nouveau problème, les forces de réactions $\underline{F}_{[p]}$ sont supposées connues et le déplacement imposé $\underline{U}_{[p]}^d$ est supposé inconnu. Le problème à résoudre est :

Trouver $\underline{U}_{[p]}^d$ tel que $\forall \underline{V}_{[p]}$, on ait

$$\begin{bmatrix} 0_{[l]}^T & \underline{V}_{[p]}^T \end{bmatrix} \begin{bmatrix} \mathbf{K}_{[ll]} & \mathbf{K}_{[lp]} \\ \mathbf{K}_{[lp]}^T & \mathbf{K}_{[pp]} \end{bmatrix} \begin{bmatrix} \underline{U}_{[l]} \\ \underline{U}_{[p]}^d \end{bmatrix} = \begin{bmatrix} 0_{[l]}^T & \underline{V}_{[p]}^T \end{bmatrix} \begin{bmatrix} \underline{F}_{[l]}^d \\ \underline{F}_{[p]} \end{bmatrix}$$

On a alors l'égalité :

$$\underline{F}_{[p]} = \mathbf{K}_{[lp]}^T \underline{U}_{[l]} + \mathbf{K}_{[pp]} \underline{U}_{[p]}^d$$

En sachant que le déplacement est complètement connu suite à la résolution du problème en déplacements, on peut donc calculer les forces de réactions.

VI.3.4 Système matriciel complet

En combinant les deux systèmes d'équations précédents :

$$\mathbf{K}_{[ll]} \underline{U}_{[l]} + \mathbf{K}_{[lp]} \underline{U}_{[p]}^d = \underline{F}_{[l]}^d$$

et

$$\mathbf{K}_{[lp]}^T \underline{U}_{[l]} + \mathbf{K}_{[pp]} \underline{U}_{[p]}^d = \underline{F}_{[p]}$$

on peut écrire le système matriciel suivant :

$$\begin{bmatrix} \mathbf{K}_{[ll]} & \mathbf{K}_{[lp]} \\ \mathbf{K}_{[lp]}^T & \mathbf{K}_{[pp]} \end{bmatrix} \begin{bmatrix} \underline{U}_{[l]} \\ \underline{U}_{[p]}^d \end{bmatrix} = \begin{bmatrix} \underline{F}_{[l]}^d \\ \underline{F}_{[p]} \end{bmatrix}$$

où les inconnues sont $\underline{U}_{[l]}$, les déplacements libres, et $\underline{F}_{[p]}$ les réactions aux appuis. Ce système est résolu dans la pratique en deux temps : calcul de $\underline{U}_{[l]}$ puis calcul de $\underline{F}_{[p]}$.

VI.3.5 Blocage des mouvements de solides rigides

On rappelle que dans ce chapitre de cours, le problème de mécanique que l'on cherche à résoudre est un problème de statique. Le système en équilibre ne doit donc pas avoir de mobilité, sinon cela devient un problème de dynamique. Autrement dit il est nécessaire que tous les mouvements de corps rigide soient bloqués. Pour un calcul bidimensionnel, il faut bloquer les 2 translations et la rotation (Fig. VI.3). Pour un calcul tridimensionnel, il faut bloquer les 3 translations et les 3 rotations.

Une mobilité se traduit mathématiquement par une singularité de la matrice $\mathbf{K}_{[ll]}$, alors que si tous les mouvements de corps rigide sont bloqués, $\mathbf{K}_{[ll]}$ est inversible.

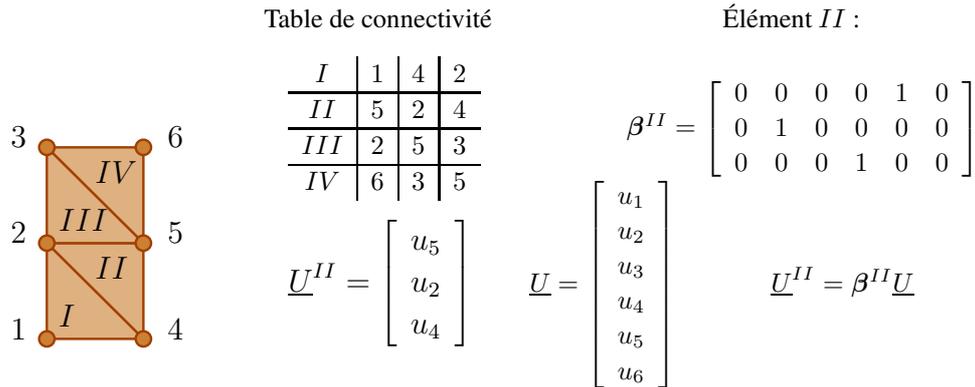


Figure VI.4 – Exemple d'utilisation de la matrice de localisation dans le cas d'un maillage de triangles à 3 nœuds.

Preuve Les déplacements de solide rigide $\underline{W} \in \mathcal{V}^h$ de la structure sont ceux pour lesquels l'énergie de déformation est nulle, soit

$$\mathcal{E}_D(\vec{w}) = \frac{1}{2} \underline{W}^T \mathbf{K} \underline{W} = 0 \quad \text{avec} \quad \underline{W} \neq \underline{0}.$$

Cela veut aussi dire que les forces nécessaires pour déplacer la structure sont nulles, donc $\underline{F} = \underline{0}$ (problème de statique, il n'y a pas d'inertie), et on a aussi :

$$\mathbf{K} \underline{W} = \underline{0} \quad \text{avec} \quad \underline{W} \neq \underline{0}.$$

Les modes de déplacements de solide rigide correspondent alors au vecteurs propres de \mathbf{K} associés à des valeurs propres nulles. •

VI.4 Construction du problème discrétisé

VI.4.1 Calcul de la matrice de rigidité

Le problème discrétisé revient à chercher la solution d'un système linéaire d'équations faisant intervenir la matrice de rigidité \mathbf{K} telle que :

$$\mathbf{K} = \int_{\Omega} \mathbf{B}^T \mathbf{C} \mathbf{B} \, d\Omega$$

Dans la pratique, cette matrice est calculée comme la somme des contributions de chaque élément :

$$\mathbf{K} = \sum_{e=1}^{\text{nb. élts}} \int_{\Omega_e} \mathbf{B}^T \mathbf{C} \mathbf{B} \, d\Omega_e$$

où Ω_e représente le domaine associé à l'élément e . Étant donné que le déplacement discrétisé $\vec{u}^h \in \mathcal{V}^h$ d'un point situé à l'intérieur d'un élément e ne fait intervenir que les fonctions de forme de cet élément, on peut calculer pour chaque élément sa contribution à \mathbf{K} indépendamment des autres éléments.

On introduit pour chaque élément une matrice de localisation β^e , créée à partir de la table de connectivité, qui permet d'extraire les déplacements \underline{U}^e des nœuds connectés à l'élément e des déplacements \underline{U} des nœuds de la structure :

$$\underline{U}^e = \beta^e \underline{U}.$$

La figure VI.4 montre l'exemple de matrices de localisation pour un maillage de triangles à 3 nœuds.

Le vecteur \underline{U}^e est la représentation du champ de déplacement \vec{u}^h dans l'élément e , dans la base des fonctions de forme Φ^e de l'élément :

$$\underline{u}_e^h(M) = \Phi^e \underline{U}^e$$

L'opérateur gradient discrétisé est défini dans l'élément par :

$$\mathbf{B}^e = \mathbf{D}\Phi^e$$

Pour tout point M de l'élément, cet opérateur élémentaire \mathbf{B}^e est relié à l'opérateur \mathbf{B} par la matrice de localisation :

$$\mathbf{B} = \mathbf{B}^e \beta^e.$$

Preuve En effet, le champ de déformation associé à \vec{u}^h vaut

$$\underline{\epsilon} = \mathbf{B}\underline{U}$$

soit aussi dans l'élément e

$$\underline{\epsilon} = \mathbf{B}^e \underline{U}^e$$

En remplaçant \underline{U}^e par $\beta^e \underline{U}$, on a :

$$\underline{\epsilon} = \mathbf{B}^e \beta^e \underline{U}$$

donc $\mathbf{B} = \mathbf{B}^e \beta^e$ dans l'élément e . •

Finalement, la matrice de rigidité s'écrit

$$\mathbf{K} = \sum_{e=1}^{\text{nb. élts}} \beta^{eT} \int_{\Omega_e} \mathbf{B}^{eT} \mathbf{C} \mathbf{B}^e \, d\Omega_e \beta^e$$

En posant \mathbf{K}^e la matrice de rigidité élémentaire telle que :

$$\mathbf{K}^e = \int_{\Omega_e} \mathbf{B}^{eT} \mathbf{C} \mathbf{B}^e \, d\Omega_e$$

on a

$$\mathbf{K} = \sum_{e=1}^{\text{nb. élts}} \beta^{eT} \mathbf{K}^e \beta^e.$$

Exercice VI.1 Dans le cas d'une fonction vectorielle, définie d'un domaine tridimensionnel Ω vers \mathbb{R}^3 , et en utilisant un tétraèdre à 4 nœuds (défini par les 4 points a, b, c et d), donner Φ^e , \underline{U}^e et \mathbf{B}^e dans un élément.

Solution :

On rappelle que les expressions des fonctions de forme de cet élément s'écrivent sous la forme :

$$\varphi_a(x, y, z) = \frac{1}{6v}(\alpha_a + \beta_a x + \gamma_a y + \delta_a z)$$

$$\varphi_b(x, y, z) = \frac{1}{6v}(\alpha_b + \beta_b x + \gamma_b y + \delta_b z)$$

$$\varphi_c(x, y, z) = \frac{1}{6v}(\alpha_c + \beta_c x + \gamma_c y + \delta_c z)$$

$$\varphi_d(x, y, z) = \frac{1}{6v}(\alpha_d + \beta_d x + \gamma_d y + \delta_d z)$$

où les coefficients $\alpha_i, \beta_i, \gamma_i$ et δ_i sont calculés à partir des coordonnées des 4 nœuds de l'élément et où v est le volume de l'élément.

Le champ de déplacement \vec{u}^h s'écrit dans l'élément sous la forme :

$$\begin{aligned} \forall M(x, y, z) \in \Omega_e, \vec{u}_e^h(M) = & u_a \varphi_a(M) \vec{e}_1 + v_a \varphi_a(M) \vec{e}_2 + w_a \varphi_a(M) \vec{e}_3 \\ & + u_b \varphi_b(M) \vec{e}_1 + v_b \varphi_b(M) \vec{e}_2 + w_b \varphi_b(M) \vec{e}_3 \\ & + u_c \varphi_c(M) \vec{e}_1 + v_c \varphi_c(M) \vec{e}_2 + w_c \varphi_c(M) \vec{e}_3 \\ & + u_d \varphi_d(M) \vec{e}_1 + v_d \varphi_d(M) \vec{e}_2 + w_d \varphi_d(M) \vec{e}_3 \end{aligned}$$

En écrivant \vec{u}_e^h dans la base $(\vec{e}_1, \vec{e}_2, \vec{e}_3)$ de \mathbb{R}^3 , on a :

$$\underline{u}_e^h(M) = \Phi^e \underline{U}^e$$

où Φ^e est la matrice des fonctions de forme de l'élément donnée par

$$\Phi^e = \begin{bmatrix} \varphi_a & 0 & 0 & \varphi_b & 0 & 0 & \varphi_c & 0 & 0 & \varphi_d & 0 & 0 \\ 0 & \varphi_a & 0 & 0 & \varphi_b & 0 & 0 & \varphi_c & 0 & 0 & \varphi_d & 0 \\ 0 & 0 & \varphi_a & 0 & 0 & \varphi_b & 0 & 0 & \varphi_c & 0 & 0 & \varphi_d \end{bmatrix}$$

et \underline{U}^e est le champ de déplacement \vec{u}^h restreint à l'élément e et écrit dans la base des fonctions de forme :

$$\underline{U}^e = [u_a \ v_a \ w_a \ u_b \ v_b \ w_b \ u_c \ v_c \ w_c \ u_d \ v_d \ w_d]^T$$

La matrice Φ^e est une matrice 3×12 , et le vecteur \underline{U}^e est une matrice colonne à 12 lignes.

L'opérateur gradient discrétisé est défini dans un élément comme une matrice 6×12 :

$$B^e = D\Phi^e = \begin{bmatrix} \frac{\partial \varphi_a}{\partial x} & 0 & 0 & \dots & \frac{\partial \varphi_d}{\partial x} & 0 & 0 \\ 0 & \frac{\partial \varphi_a}{\partial y} & 0 & \dots & 0 & \frac{\partial \varphi_d}{\partial y} & 0 \\ 0 & 0 & \frac{\partial \varphi_a}{\partial z} & \dots & 0 & 0 & \frac{\partial \varphi_d}{\partial z} \\ 0 & \frac{\partial \varphi_a}{\partial z} & \frac{\partial \varphi_a}{\partial y} & \dots & 0 & \frac{\partial \varphi_d}{\partial z} & \frac{\partial \varphi_d}{\partial y} \\ \frac{\partial \varphi_a}{\partial z} & 0 & \frac{\partial \varphi_a}{\partial x} & \dots & \frac{\partial \varphi_d}{\partial z} & 0 & \frac{\partial \varphi_d}{\partial x} \\ \frac{\partial \varphi_a}{\partial y} & \frac{\partial \varphi_a}{\partial x} & 0 & \dots & \frac{\partial \varphi_d}{\partial y} & \frac{\partial \varphi_d}{\partial x} & 0 \end{bmatrix}$$

En calculant les dérivées des fonctions de forme, on obtient :

$$B^e = \frac{1}{6v} \begin{bmatrix} \beta_a & 0 & 0 & \beta_b & 0 & 0 & \beta_c & 0 & 0 & \beta_d & 0 & 0 \\ 0 & \gamma_a & 0 & 0 & \gamma_b & 0 & 0 & \gamma_c & 0 & 0 & \gamma_d & 0 \\ 0 & 0 & \delta_a & 0 & 0 & \delta_b & 0 & 0 & \delta_c & 0 & 0 & \delta_d \\ 0 & \delta_a & \gamma_a & 0 & \delta_b & \gamma_b & 0 & \delta_c & \gamma_c & 0 & \delta_d & \gamma_d \\ \delta_a & 0 & \beta_a & \delta_b & 0 & \beta_b & \delta_c & 0 & \beta_c & \delta_d & 0 & \beta_d \\ \gamma_a & \beta_a & 0 & \gamma_b & \beta_b & 0 & \gamma_c & \beta_c & 0 & \gamma_d & \beta_d & 0 \end{bmatrix}$$

Dans ce cas particulier du tétraèdre à 4 nœuds, étant donné que la matrice $B^e(M)$ est constante dans l'élément, le calcul de K^e est immédiat (v est le volume de l'élément) :

$$K^e = v B^{eT} C^e B^e.$$

VI.4.2 Calcul des forces aux nœuds

Le second membre du système linéaire à résoudre est le vecteur des forces aux nœuds donné par :

$$\underline{F} = \underbrace{\int_{\Omega} \Phi^T \underline{f} \, d\Omega}_{\text{terme A}} + \underbrace{\int_{\partial_F \Omega} \Phi^T \underline{F}^d \, dS}_{\text{terme B}}$$

Le terme A agit sur tous les éléments subissant une force volumique (Fig. VI.5). Le terme B agit sur les éléments dont une frontière est soumise à une force imposée (Fig. VI.5).

En utilisant la matrice de localisation précédente, on a :

$$\underline{F} = \sum_{e=1}^{n_A} \beta^{eT} \int_{\Omega_e} \Phi^{eT} \underline{f} \, d\Omega_e + \sum_{e=1}^{n_B} \beta^{eT} \int_{\partial_F \Omega_e} \Phi^{eT} \underline{F}^d \, dS$$

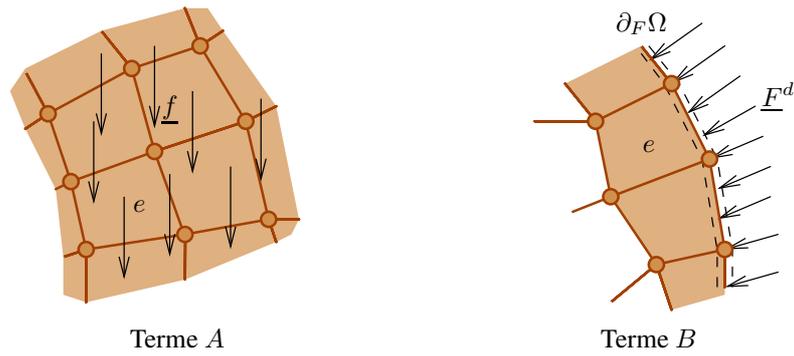


Figure VI.5 – Forces agissant sur les éléments.

où n_A est le nombre d'éléments dans le volume, et n_B est le nombre d'éléments touchant la surface $\partial_F \Omega$. Le vecteur des forces aux nœuds est donc calculé comme une somme de contributions de chaque élément.

VII — Calculs au niveau élémentaire

VII.1 Élément de référence

VII.1.1 Nécessité d'un élément de référence

Les éléments étudiés jusqu'à présent (barre à 2 nœuds en 1D, triangle à 3 nœuds en 2D et tétraèdre à 4 nœuds en 3D) sont construits directement dans l'espace physique. Ces trois éléments peuvent être construits ainsi, mais la plupart des autres éléments font appel à un élément de référence, autrement les éléments ne peuvent pas s'assembler les uns aux autres.

Prenons l'exemple d'un quadrangle à 4 nœuds, dont la base d'approximation est $[1 \ x \ y \ xy]$, construit sur les 4 nœuds suivants : $(0, 0)$, $(1, 0)$, $(1, 1 + \alpha)$ et $(0, 1)$ (Fig. VII.1).

Le système à résoudre pour trouver les fonctions de forme est :

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 + \alpha & 1 \\ 0 & 0 & 1 + \alpha & 0 \end{bmatrix} \begin{bmatrix} \varphi_a(x, y) \\ \varphi_b(x, y) \\ \varphi_c(x, y) \\ \varphi_d(x, y) \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ y \\ xy \end{bmatrix}$$

La solution est

$$\begin{aligned} \varphi_a(x, y) &= (1 - x)(1 - y) \ ; \ \varphi_b(x, y) = x\left(1 - \frac{y}{1 + \alpha}\right) \ ; \\ \varphi_c(x, y) &= \frac{xy}{1 + \alpha} \ ; \ \varphi_d(x, y) = y(1 - x) \ ; \end{aligned}$$

Les fonctions de forme sont tracées sur la figure VII.1. La fonction φ_a devrait s'annuler sur les bords opposés au nœud a (bords $c - d$ et $b - c$) afin d'éviter que ce nœud ait une influence hors de l'élément, mais φ_a ne vaut pas 0 sur le bord $c - d$. La même conclusion apparaît pour la

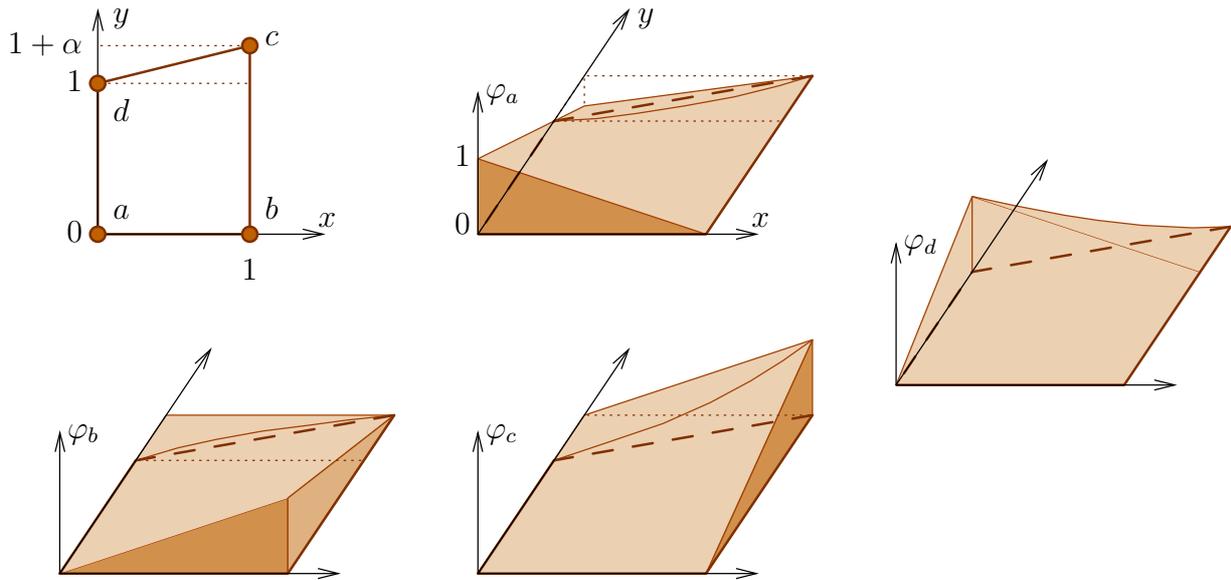


Figure VII.1 – Elément quadrangle dont les fonctions de forme ne s'annulent pas sur les bords.

fonction φ_b . Cet élément ne peut pas s'assembler avec d'autres le long du bord $c-d$: les fonctions de forme ne seraient pas continues. On construit alors les fonctions de forme dans un élément de référence de forme géométrique simple, puis une transformation géométrique permet le passage de l'élément de référence à l'élément physique.

VII.1.2 Géométrie de l'élément de référence

Un élément de référence est un élément de forme simple qui permet de construire les fonctions de forme. Un système de coordonnées lui est attaché, généralement compris entre -1 et $+1$ dans chaque direction. Une transformation permet de passer de l'élément de référence à l'élément physique.

Plusieurs exemples d'éléments de référence sont décrits sur la figure VII.2.

VII.1.3 Fonctions de forme un élément de référence à 2 variables

On se place dans le cas des fonctions à 2 variables définies de l'élément du plan Ω_e vers \mathbb{R} . On utilise l'élément quadrangle à 4 nœuds dans la suite pour illustrer le principe (Fig. VII.3). L'espace vectoriel \mathcal{V}_e^h engendré par les 4 fonctions de forme associées aux 4 nœuds a, b, c et d est de dimension 4.

On définit un système de coordonnées de référence (r, s) tel que sur les bords de l'élément on ait $r = \pm 1$ et $s = \pm 1$. Le carré défini dans le repère (r, s) dont les sommets ont pour coordonnées $(-1, -1)$ et $(1, 1)$ est appelé élément de référence. Les coordonnées d'un nœud i dans l'élément de référence sont notées (r_i, s_i) .

La base canonique polynomiale de \mathcal{V}_e^h , de dimension 4, doit contenir 4 termes. Pour le quadrangle à 4 nœuds, on choisit $[1 \ r \ s \ rs]$ dans l'élément de référence. Cette base est symétrique, on peut échanger r et s sans changer la base. Le passage de la base canonique à la base des

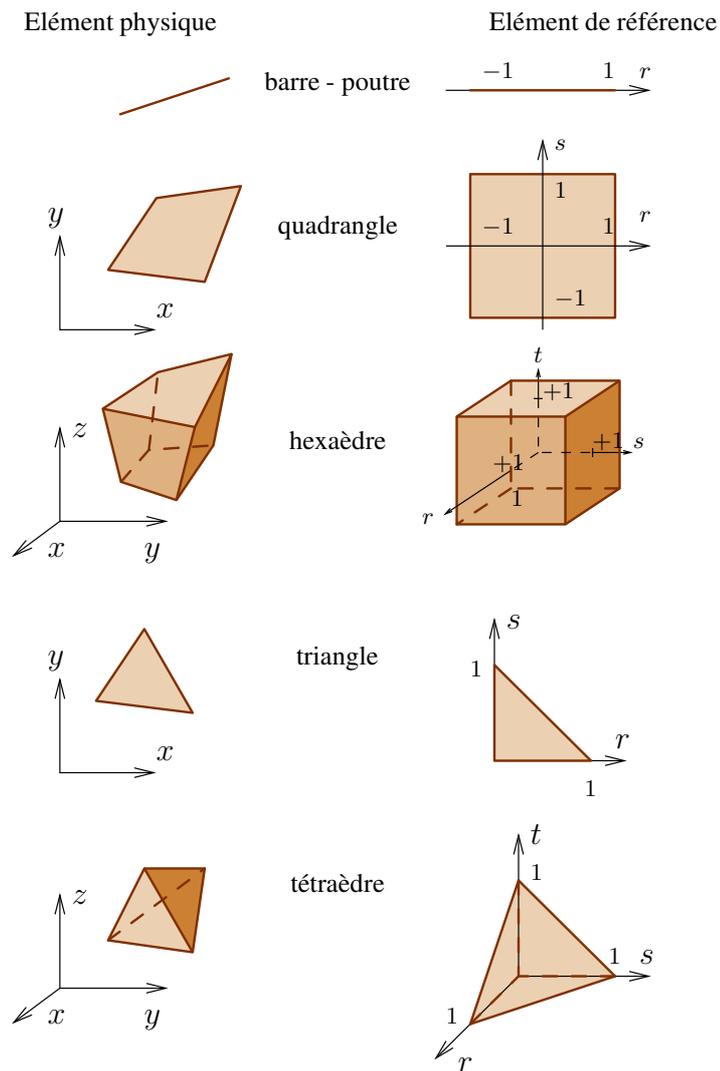


Figure VII.2 – Quelques éléments physiques et éléments de références associés.

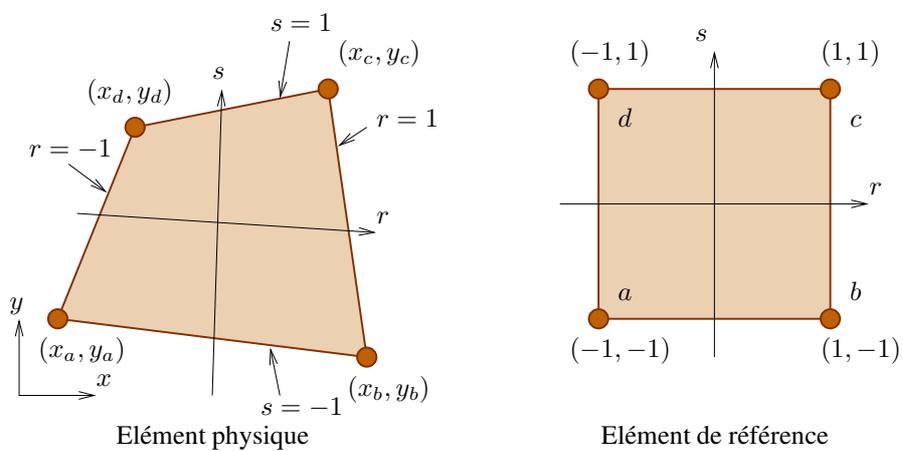


Figure VII.3 – Passage de l'élément physique à l'élément de référence pour un quadrangle.

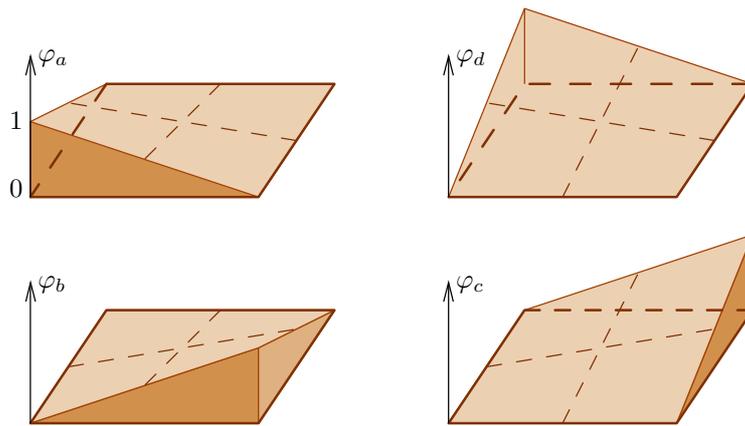


Figure VII.4 – Fonctions de forme du quadrangle à 4 nœuds dans l'élément de référence.

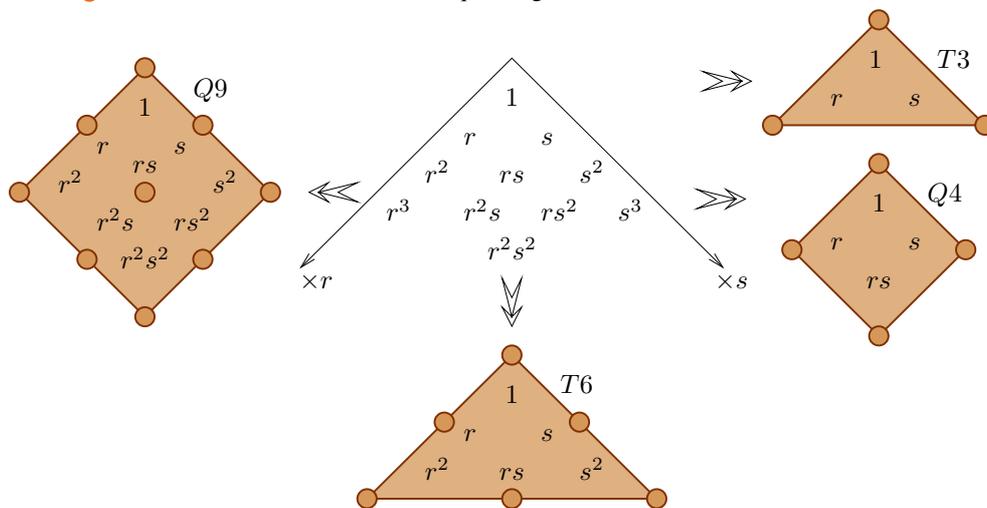


Figure VII.5 – Triangle de Pascal.

fonctions de forme revient à résoudre le système suivant :

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ r_a & r_b & r_c & r_d \\ s_a & s_b & s_c & s_d \\ r_a s_a & r_b s_b & r_c s_c & r_d s_d \end{bmatrix} \begin{bmatrix} \varphi_a \\ \varphi_b \\ \varphi_c \\ \varphi_d \end{bmatrix} = \begin{bmatrix} 1 \\ r \\ s \\ rs \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} \varphi_a \\ \varphi_b \\ \varphi_c \\ \varphi_d \end{bmatrix} = \begin{bmatrix} 1 \\ r \\ s \\ rs \end{bmatrix}$$

La résolution du système se fait par des opérations sur les lignes :

$$(L_1) - (L_2) - (L_3) + (L_4) \Rightarrow 4\varphi_a = 1 - r - s + rs$$

$$(L_1) + (L_2) - (L_3) - (L_4) \Rightarrow 4\varphi_b = 1 + r - s - rs$$

$$(L_1) + (L_2) + (L_3) + (L_4) \Rightarrow 4\varphi_c = 1 + r + s + rs$$

$$(L_1) - (L_2) + (L_3) - (L_4) \Rightarrow 4\varphi_d = 1 - r + s - rs$$

La solution est

$$\begin{aligned} \varphi_a(r, s) &= \frac{1}{4}(1-r)(1-s), & \varphi_b(r, s) &= \frac{1}{4}(1+r)(1-s), \\ \varphi_c(r, s) &= \frac{1}{4}(1+r)(1+s), & \varphi_d(r, s) &= \frac{1}{4}(1-r)(1+s). \end{aligned}$$

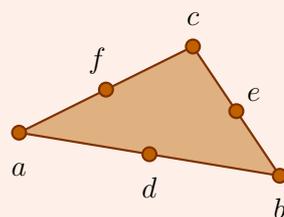
Les fonctions φ_i peuvent s'écrire aussi plus simplement :

$$\varphi_i(r, s) = \frac{1}{4}(1 + r_i r)(1 + s_i s).$$

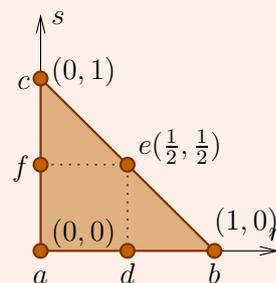
Ces fonctions de formes sont tracées sur la figure VII.4

Le triangle de Pascal explicité sur la figure VII.5 permet de trouver la base canonique polynômiale à 2 variables associée à d'autres éléments du plan. On écrit 1 sur la ligne du haut, on complète la ligne du dessous en multipliant par r vers la gauche et par s vers la droite chacun des termes de la ligne du dessus. Plusieurs éléments classiquement utilisés sont présentés sur cette figure, avec leurs base canoniques respectives.

Exercice VII.1 Construire les fonctions de forme du triangle à 6 nœuds dans l'élément de référence. Tracer ces 6 fonctions.



Élément physique



Élément de référence

VII.1.4 Passage de l'élément de référence à l'élément physique

Cas général

Les coordonnées d'un point quelconque situé dans l'élément physique (x, y, z) (ou (x, y) dans le plan) sont reliées aux coordonnées dans l'élément de référence par les fonctions de forme et aux coordonnées des nœuds de l'élément.

En notant \underline{x}^e les coordonnées physiques d'un point M de l'élément e et \underline{X}^e les coordonnées physiques des nœuds de l'élément e , on peut écrire

$$\underline{x}^e = \Phi^e \underline{X}^e$$

où Φ^e est la matrice des fonctions de forme. Un tel élément est dit isoparamétrique car il fait intervenir la même approximation pour la géométrie (passage de l'élément de référence à l'élément physique) et pour le déplacement.

On définit le gradient \mathbf{F} de la transformation permettant de passer de l'élément de référence à l'élément physique dans le cas général en 3 dimensions comme :

$$\text{en 3 dimensions : } \mathbf{F} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial s} & \frac{\partial x}{\partial t} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial s} & \frac{\partial y}{\partial t} \\ \frac{\partial z}{\partial r} & \frac{\partial z}{\partial s} & \frac{\partial z}{\partial t} \end{bmatrix} ; \text{ en 2 dimensions : } \mathbf{F} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial s} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial s} \end{bmatrix}$$

Application au quadrangle à 4 nœuds

Par exemple, pour l'élément quadrangle à 4 nœuds du plan, on a

$$\underline{x}^e = \begin{bmatrix} \varphi_a(r, s) & 0 & \varphi_b(r, s) & 0 & \varphi_c(r, s) & 0 & \varphi_d(r, s) & 0 \\ 0 & \varphi_a(r, s) & 0 & \varphi_b(r, s) & 0 & \varphi_c(r, s) & 0 & \varphi_d(r, s) \end{bmatrix} \begin{bmatrix} x_a \\ y_a \\ x_b \\ y_b \\ x_c \\ y_c \\ x_d \\ y_d \end{bmatrix}$$

ou encore

$$x = \sum_{i=a}^d \varphi_i(r, s) x_i, \quad y = \sum_{i=a}^d \varphi_i(r, s) y_i.$$

Le calcul de $\frac{\partial x}{\partial r}$ et de $\frac{\partial x}{\partial s}$ s'obtient par :

$$\frac{\partial x}{\partial r} = \begin{bmatrix} \varphi_{a,r} & \varphi_{b,r} & \varphi_{c,r} & \varphi_{d,r} \end{bmatrix} \begin{bmatrix} x_a \\ x_b \\ x_c \\ x_d \end{bmatrix}, \quad \frac{\partial x}{\partial s} = \begin{bmatrix} \varphi_{a,s} & \varphi_{b,s} & \varphi_{c,s} & \varphi_{d,s} \end{bmatrix} \begin{bmatrix} x_a \\ x_b \\ x_c \\ x_d \end{bmatrix}.$$

Le calcul de $\frac{\partial y}{\partial r}$ et de $\frac{\partial y}{\partial s}$ s'obtient par :

$$\frac{\partial y}{\partial r} = \begin{bmatrix} \varphi_{a,r} & \varphi_{b,r} & \varphi_{c,r} & \varphi_{d,r} \end{bmatrix} \begin{bmatrix} y_a \\ y_b \\ y_c \\ y_d \end{bmatrix}, \quad \frac{\partial y}{\partial s} = \begin{bmatrix} \varphi_{a,s} & \varphi_{b,s} & \varphi_{c,s} & \varphi_{d,s} \end{bmatrix} \begin{bmatrix} y_a \\ y_b \\ y_c \\ y_d \end{bmatrix}.$$

VII.2 Matrice de rigidité dans l'élément de référence

VII.2.1 Expression

Le calcul de la matrice de rigidité \mathbf{K}^e fait intervenir une intégrale dans l'élément physique. Cette intégrale est généralement calculée dans l'élément de référence pour deux raisons :

- les fonctions de forme sont connues dans l'élément de référence,
- l'élément physique a une forme souvent compliquée, ce qui rend l'intégration directe dans l'élément physique difficile.

L'expression de la matrice de rigidité d'un élément est

$$\mathbf{K}^e = \int_{\Omega_e} \mathbf{B}^{eT}(M) \mathbf{C}^e \mathbf{B}^e(M) d\Omega$$

où Ω_e est l'élément physique. Cette intégrale s'écrit dans l'élément de référence en faisant intervenir le déterminant du gradient \mathbf{F} de la transformation permettant de passer de l'élément de référence à l'élément physique :

$$\mathbf{K}^e = \int_{\Omega_{ref}} \mathbf{B}^{eT}(r, s) \mathbf{C}^e \mathbf{B}^e(r, s) \det \mathbf{F} d\Omega_{ref}$$

où Ω_{ref} est l'élément de référence.

VII.2.2 Cas du quadrangle

Afin d'illustrer le principe, on se place dans le cas de l'élasticité plane et avec une discrétisation utilisant des quadrangles à 4 nœuds. Le champ de déplacement \underline{u}_e^h dans un tel élément s'écrit :

$$\underline{u}_e^h = \begin{bmatrix} \varphi_a(r, s) & 0 & \varphi_b(r, s) & 0 & \varphi_c(r, s) & 0 & \varphi_d(r, s) & 0 \\ 0 & \varphi_a(r, s) & 0 & \varphi_b(r, s) & 0 & \varphi_c(r, s) & 0 & \varphi_d(r, s) \end{bmatrix} \begin{bmatrix} u_a \\ v_a \\ u_b \\ v_b \\ u_c \\ v_c \\ u_d \\ v_d \end{bmatrix}$$

$$\underline{u}_e^h = \Phi^e(r, s) \underline{U}^e$$

où $u_i \vec{e}_1 + v_i \vec{e}_2$ est le déplacement du nœud i .

Le champ de déformation associé à \underline{u}_e^h vaut :

$$\underline{\epsilon}^e(\underline{u}_e^h) = \mathbf{D} \Phi^e(r, s) \underline{U}^e = \mathbf{B}^e(r, s) \underline{U}^e$$

avec

$$\mathbf{D} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} .$$

L'opérateur gradient discrétisé vaut alors :

$$\mathbf{B}^e(r, s) = \begin{bmatrix} \varphi_{a,x} & 0 & \varphi_{b,x} & 0 & \varphi_{c,x} & 0 & \varphi_{d,x} & 0 \\ 0 & \varphi_{a,y} & 0 & \varphi_{b,y} & 0 & \varphi_{c,y} & 0 & \varphi_{d,y} \\ \varphi_{a,y} & \varphi_{a,x} & \varphi_{b,y} & \varphi_{b,x} & \varphi_{c,y} & \varphi_{c,x} & \varphi_{d,y} & \varphi_{d,x} \end{bmatrix}.$$

Or $\varphi_i(r, s)$ est une fonction de r et s , dont les dérivées par rapport à x et y valent

$$\frac{\partial \varphi_i}{\partial x} = \frac{\partial \varphi_i}{\partial r} \frac{\partial r}{\partial x} + \frac{\partial \varphi_i}{\partial s} \frac{\partial s}{\partial x} \quad \text{et} \quad \frac{\partial \varphi_i}{\partial y} = \frac{\partial \varphi_i}{\partial r} \frac{\partial r}{\partial y} + \frac{\partial \varphi_i}{\partial s} \frac{\partial s}{\partial y}$$

ce qui s'écrit sous forme matricielle,

$$\begin{bmatrix} \frac{\partial \varphi_i}{\partial x} \\ \frac{\partial \varphi_i}{\partial y} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial s}{\partial x} \\ \frac{\partial r}{\partial y} & \frac{\partial s}{\partial y} \end{bmatrix}}_{=\mathbf{F}^{-T}} \begin{bmatrix} \frac{\partial \varphi_i}{\partial r} \\ \frac{\partial \varphi_i}{\partial s} \end{bmatrix}$$

où \mathbf{F} est le gradient de la transformation passant d'un point (r, s) de l'élément de référence à son "image" (x, y) dans l'élément physique. Les dérivées des fonctions $\varphi_i(r, s)$ par rapport à r et s sont facilement calculables de façon analytique. On a pour le quadrangle à 4 nœuds :

$$\begin{aligned} \varphi_{a,r} &= -\frac{1}{4}(1-s) \quad , \quad \varphi_{a,s} = -\frac{1}{4}(1-r) \quad , \quad \varphi_{b,r} = \frac{1}{4}(1-s) \quad , \quad \varphi_{b,s} = -\frac{1}{4}(1+r) \quad , \\ \varphi_{c,r} &= \frac{1}{4}(1+s) \quad , \quad \varphi_{c,s} = \frac{1}{4}(1+r) \quad , \quad \varphi_{d,r} = -\frac{1}{4}(1+s) \quad , \quad \varphi_{d,s} = \frac{1}{4}(1-r) \quad . \end{aligned}$$

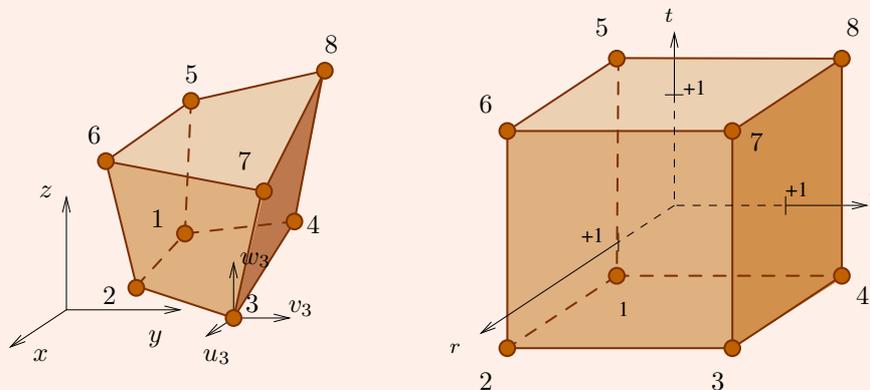
Le calcul de la transformation pour passer de l'élément de référence à l'élément physique ainsi que le calcul de \mathbf{F} sont détaillés dans le paragraphe précédent pour le quadrangle à 4 nœuds.

L'expression de la matrice de rigidité du quadrangle à 4 nœuds est

$$\mathbf{K}^e = \int_{\Omega^e} \mathbf{B}^{eT}(M) \mathbf{C}^e \mathbf{B}^e(M) d\Omega = \int_{-1}^1 \int_{-1}^1 h^e \mathbf{B}^{eT}(r, s) \mathbf{C}^e \mathbf{B}^e(r, s) \det \mathbf{F} dr ds$$

où h^e est l'épaisseur de l'élément e .

Exercice VII.2 Construire les fonctions de forme du cube à 8 nœuds dans l'élément de référence. Reprendre la démarche précédente en l'appliquant au cas tridimensionnel de l'héxaèdre à 8 nœuds.



VIII — Intégration numérique

VIII.1 Intégration numérique de la matrice de rigidité

La matrice de rigidité élémentaire peut toujours s'écrire comme une intégrale dans l'élément de référence Ω_{ref} :

$$\mathbf{K}^e = \int_{\Omega_{ref}} \mathbf{B}^{eT} \mathbf{C}^e \mathbf{B}^e \det \mathbf{F} d\Omega$$

Ce calcul n'est en général pas réalisable de façon analytique, on calcule alors numériquement une approximation de cette intégrale. L'idée est de transformer l'intégrale en une somme de termes évalués en des points précis (points d'intégration) de l'élément de référence, pondérés par des coefficients (poids). En notant G le nombre de points d'intégration, l'approximation de la matrice de rigidité élémentaire vaut

$$\mathbf{K}^e = \sum_{g=1}^G \mathbf{B}^{eT}|_g \mathbf{C}^e \mathbf{B}^e|_g \det \mathbf{F}|_g w_g$$

où la notation $|_g$ veut dire que la quantité est évaluée au point d'intégration g et w_g est le poids associé à ce point.

L'algorithme de calcul de \mathbf{K}^e par intégration numérique est alors :

1. Initialiser \mathbf{K}^e à zéro
2. Pour chaque point d'intégration g faire :
 - (a) Déterminer les coordonnées du point dans l'élément de référence et son poids w_g
 - (b) Calculer les dérivées des fonctions de forme φ_i par rapport aux coordonnées de référence au point d'intégration g

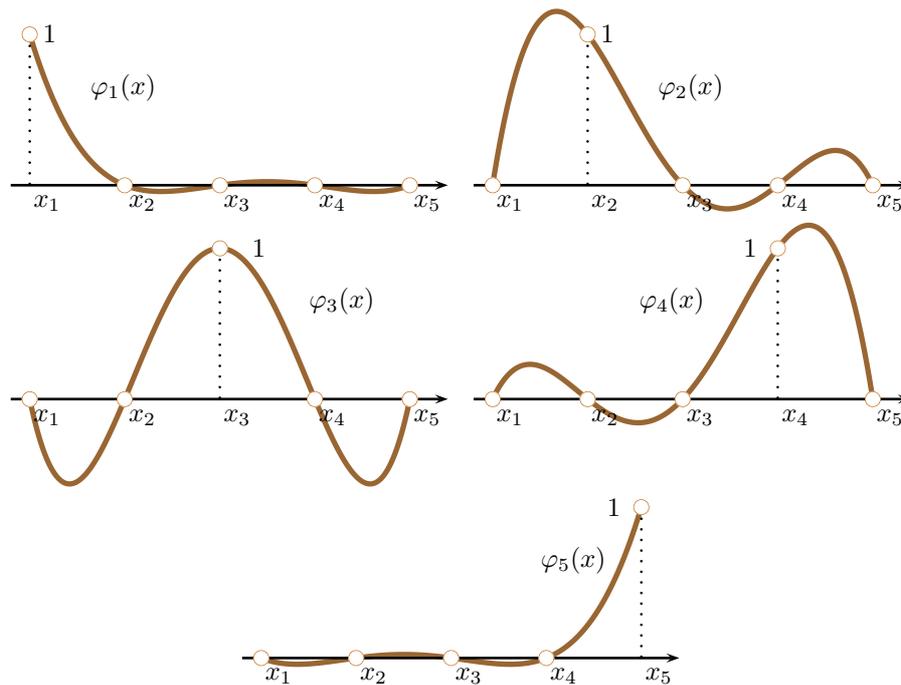


Figure VIII.1 – Polynômes de Lagrange de degrés 4 associés à 5 points.

- (c) Calculer $\mathbf{F}|_g$, en déduire $\mathbf{F}^{-1}|_g$ et $\det \mathbf{F}$
- (d) Calculer les dérivées des fonctions de forme φ_i par rapport aux coordonnées physique au point d'intégration g
- (e) Evaluer $\mathbf{B}|_g$
- (f) Evaluer la contribution \mathbf{k}_g du point d'intégration à \mathbf{K}^e

$$\mathbf{k}_g = \mathbf{B}^T|_g \mathbf{C} \mathbf{B}|_g \det \mathbf{F}|_g w_g$$

- (g) Ajouter \mathbf{k}_g à \mathbf{K}^e :

$$\mathbf{K}^e = \mathbf{K}^e + \mathbf{k}_g$$

Le calcul des forces extérieures se fait aussi par intégration numérique en utilisant la même stratégie d'intégration.

Dans la suite du cours, on examine les positions et les poids d'intégration pour les éléments à 1, 2 et 3 variables.

VIII.2 Intégration des fonctions à une variable

VIII.2.1 Polynômes de Lagrange

On considère l'ensemble des fonctions à une variable définies de $[A, B]$ vers \mathbb{R} :

$$\mathcal{V} = \left\{ \begin{array}{l} v \text{ "régulier"} : [A, B] \rightarrow \mathbb{R} \\ x \mapsto v(x) \end{array} \right\}$$

L'objectif est de construire un sous-espace vectoriel \mathcal{V}^h de dimension finie n de \mathcal{V} . Pour construire ce sous-espace, on peut se donner n fonctions polynômiales $\varphi_i(x) \in \mathcal{V}$ avec $i \in [1, 2, \dots, n]$. Ces

n fonctions engendrent un sous-espace vectoriel de \mathcal{V} . Si elles sont linéairement indépendantes, elles forment alors une base de \mathcal{V}^h . On introduit les n points $x_i \in [A, B]$ distincts tels que $x_1 = A$ et $x_n = B$. On cherche à construire ces fonctions sur $[A, B]$ comme des polynômes de degré $n - 1$ en considérant $[A, B]$ comme un seul élément. Ces fonctions sont construites de manière à avoir :

- $\varphi_i(x)$ s'annule en tous les points x_j sauf en x_i ,
- $\varphi_i(x)$ vaut en au point x_i .

Autrement dit, on cherche des polynômes de degrés $n - 1$ tels que $\varphi_i(x_j) = \delta_{ij}$ où δ_{ij} est le symbole de Kronecker. On peut définir \mathcal{V}^h comme

$$\mathcal{V}^h = \text{vect} \{1, x, x^2, \dots, x^{n-1}\} = \text{vect} \{\varphi_1(x), \varphi_2(x), \dots, \varphi_n(x)\}$$

Les fonctions φ_i sont alors définies comme des polynômes de Lagrange (Fig. VIII.1) puisque l'on connaît leurs zéros :

$$\varphi_i(x) = \frac{\prod_{j=1, j \neq i}^n (x - x_j)}{\prod_{j=1, j \neq i}^n (x_i - x_j)}.$$

Le numérateur annule φ_i aux points x_j avec $i \neq j$ tandis que le dénominateur permet d'avoir $\varphi_i(x_i) = 1$ (sans sommation implicite). On note P_i^{n-1} ce polynôme de Lagrange associé au point i .

On peut aussi les construire comme la solution du système d'équations permettant de passer de la base canonique polynomiale à la base des polynômes de Lagrange (voir le cours sur la construction des fonctions de forme) :

$$\underbrace{\begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \dots & x_n^{n-1} \end{bmatrix}}_{=A} \begin{bmatrix} \varphi_1(x) \\ \varphi_2(x) \\ \varphi_3(x) \\ \vdots \\ \varphi_n(x) \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^{n-1} \end{bmatrix}$$

La solution de ce système donne évidemment les mêmes polynômes de Lagrange φ_i , mais la résolution est moins aisée.

Exercice VIII.1 Trouver les polynômes de Lagrange associés aux 2 points 0 et l .

Solution :

$$\varphi_i(x) = \frac{\prod_{j=1, j \neq i}^n (x - x_j)}{\prod_{j=1, j \neq i}^n (x_i - x_j)}$$

avec $n = 2$, $x_1 = 0$, $x_2 = l$ et i qui prend les valeurs 1 et 2, on a :

$$\varphi_1(x) = \frac{x - x_2}{x_1 - x_2} = \frac{x - l}{0 - l} = 1 - \frac{x}{l}$$

$$\varphi_2(x) = \frac{x - x_1}{x_2 - x_1} = \frac{x - 0}{l - 0} = \frac{x}{l}$$

Exercice VIII.2 Trouver les polynômes de Lagrange associés aux 3 points -1 , 0 et 1 . Ces polynômes sont dessinés sur la figure VIII.2. ■

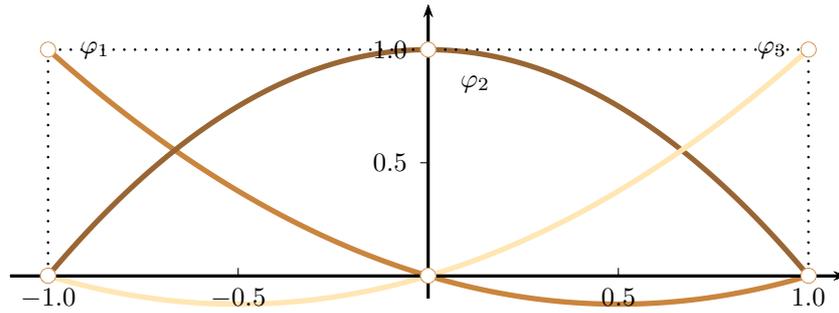


Figure VIII.2 – Polynômes de Lagrange de degrés 2 associés à 3 points.

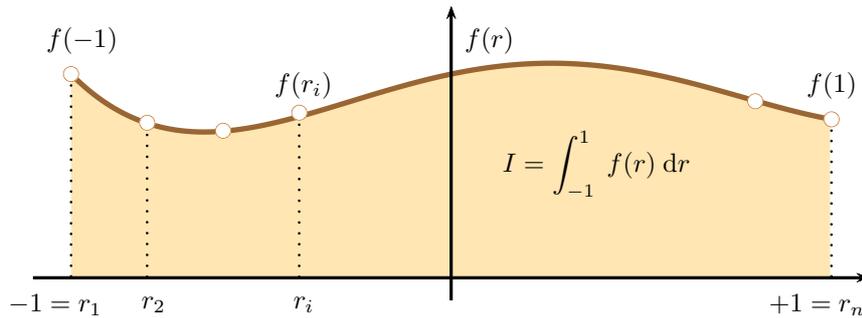


Figure VIII.3 – Intégration par la méthode de Newton-Côtes.

VIII.2.2 Méthode de Newton-Côtes

On souhaite calculer l'intégrale I d'une fonction $f(r)$ entre -1 et 1 (Fig. VIII.3) :

$$I = \int_{-1}^1 f(r) dr .$$

On choisit n points r_i régulièrement espacés dans l'intervalle $[-1, 1]$ avec $r_1 = -1$ et $r_n = 1$. On approxime $f(r)$ par un polynôme $\tilde{f}(r)$ de degré $n - 1$ prenant les valeurs $f(r_i)$ aux points r_i . L'expression de $\tilde{f}(r)$ est

$$\tilde{f}(r) = \sum_{i=1}^n f(r_i) P_i^{n-1}(r)$$

où $P_i^{n-1}(r)$ est le polynôme de degré $n - 1$ qui vaut 1 en r_i et 0 en $r_{j \neq i}$. En écrivant $P_i^{n-1}(r)$ comme un polynôme de Lagrange, on a

$$P_i^{n-1}(r) = \frac{\prod_{j=1, j \neq i}^n (r - r_j)}{\prod_{j=1, j \neq i}^n (r_i - r_j)} .$$

Donc l'approximation \tilde{I} de I vaut

$$\tilde{I} = \int_{-1}^1 \tilde{f}(r) dr = \sum_{i=1}^n f(r_i) \int_{-1}^1 P_i^{n-1}(r) dr$$

ou encore

$$\tilde{I} = \sum_{i=1}^n f(r_i) w_i$$

avec

$$w_i = \int_{-1}^1 P_i^{n-1}(r) dr$$

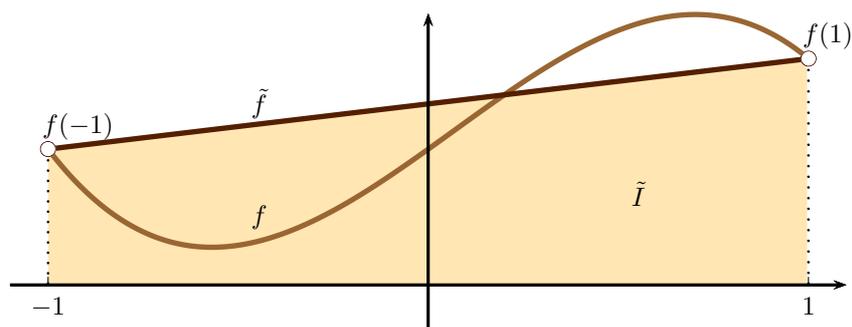


Figure VIII.4 – Méthode de Newton avec 2 points d'intégration - Formule des trapèzes

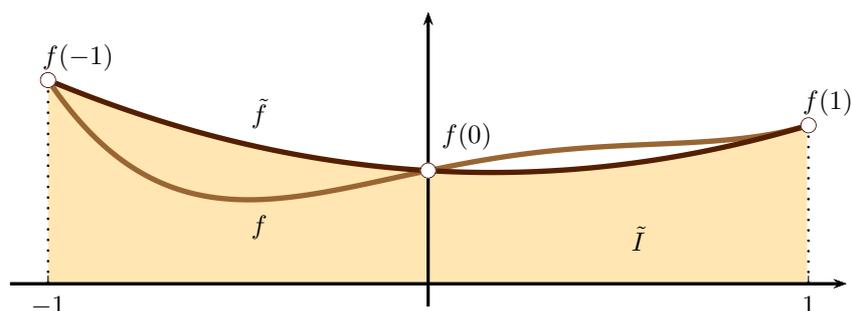


Figure VIII.5 – Méthode de Newton-Côtes avec 3 points d'intégration - Formule de Simpson

que l'on sait calculer puisque les fonctions $P_i^{n-1}(r)$ sont connues. Les nombres w_i sont appelés poids d'intégration associés aux points d'intégration r_i .

Cette méthode intègre exactement les polynômes de degré $n - 1$ avec n points d'intégration, puisque $\tilde{f}(r)$ reproduit exactement les polynômes de degré $n - 1$. Les figures VIII.4 et VIII.5 montrent de façon graphique l'intégration d'une fonction avec 2 puis 3 points d'intégration.

On remarque que la somme des poids d'intégration est toujours égale à la longueur de l'intervalle intégré, soit 2 ici. En effet, si $f(r) = 1$, alors

$$\tilde{I} = I = \int_{-1}^1 1 \, dr = 2 = \sum_{i=1}^n w_i$$

Exercice VIII.3 Trouver les poids associés à 3 points d'intégration placés en $-1, 0$ et 1 (fig. VIII.5). ■

VIII.2.3 Méthode de Gauss

En choisissant judicieusement les emplacements et les poids des points d'intégration, on peut intégrer exactement les polynômes d'ordre $2n - 1$ avec n points.

Les coordonnées et les poids des points peuvent être trouvés analytiquement. Pour illustrer la méthode, on cherche les points et leurs poids qui permettent d'intégrer exactement les polynômes d'ordre 3 avec 2 points. La fonction f à intégrer vaut

$$f(r) = a_0 + a_1 r + a_2 r^2 + a_3 r^3$$

et son intégrale entre -1 et 1 vaut

$$I = \int_{-1}^1 f(r) \, dr = 2a_0 + \frac{2}{3}a_2.$$

nb. points	degré d'intégration	r_i	w_i
2	3	$\pm 0,5773502692$	1,0000000000
3	5	$\pm 0,7745966692$ 0,0000000000	0,5555555556 0,8888888889
4	7	$\pm 0,8611363116$ $\pm 0,3399810436$	0,3478548451 0,6521451549
5	9	$\pm 0,9061798459$ $\pm 0,5384693101$ 0,0000000000	0,2369268850 0,4786286705 0,5688888889
6	11	$\pm 0,9324695142$ $\pm 0,6612093865$ $\pm 0,2386191861$	0,1713244924 0,3607615730 0,4679139346
7	13	$\pm 0,9491079123$ $\pm 0,7415311856$ $\pm 0,4058451514$ 0,0000000000	0,1294849662 0,2797053915 0,3818300505 0,4179591837

Tableau VIII.1 – Positions et poids des points d'intégration de la méthode de Gauss.

On prend 2 points d'intégration, symétriques par rapport à 0, $r_1 = -R$ et $r_2 = R$ ayant les mêmes poids $w_1 = w_2 = w$. On a alors :

$$\tilde{I} = wf(-R) + wf(R)$$

soit

$$\tilde{I} = w(2a_0 + 2a_2R^2) = 2w(a_0 + a_2R^2).$$

Pour que f soit intégrée exactement, il faut que $I = \tilde{I}$, et ce quelques soient a_0, a_1, a_2 et a_3 donc

$$I = \tilde{I} \Leftrightarrow 2a_0 + \frac{2}{3}a_2 = 2w(a_0 + a_2R^2) \Leftrightarrow w = 1 \text{ et } R^2 = \frac{1}{3}.$$

Finalement, en prenant comme points d'intégration $r_1 = -\frac{1}{\sqrt{3}}$ et $r_2 = \frac{1}{\sqrt{3}}$ avec comme poids $w_1 = w_2 = 1$, on intègre exactement tous les polynômes d'ordre 3 dans l'intervalle $[-1, 1]$.

On peut généraliser pour les polynômes d'ordre $2n - 1$, le tableau VIII.1 donne les positions et les poids des points de Gauss pour n de 2 à 7. Une intégration par points de Gauss nécessite moins de points que par la méthode de Newton-Côtes, ce qui entraîne un gain en temps de calcul.

Exercice VIII.4 Retrouver les positions et les poids pour 3 points de Gauss donnés dans le tableau VIII.1. ■

VIII.3 Intégration des fonctions à deux variables

VIII.3.1 Cas des éléments quadrangles

Dans le cas de l'élasticité plane, la matrice de rigidité élémentaire d'un quadrangle vaut :

$$\mathbf{K}^e = \int_{-1}^1 \int_{-1}^1 h^e \mathbf{B}^{eT}(r, s) \mathbf{C}^e \mathbf{B}^e(r, s) \det \mathbf{F} \, dr \, ds.$$

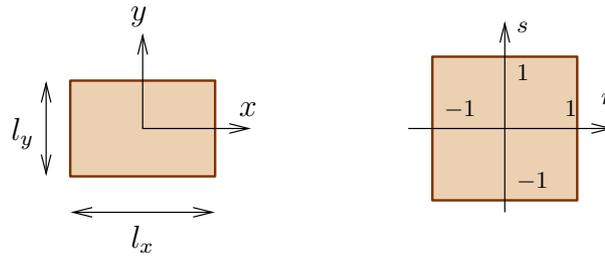


Figure VIII.6 – Éléments quadrangle physique de forme rectangulaire et élément de référence quadrangle.

où h^e est l'épaisseur dans l'élément. Le choix du nombre de points de Gauss se fait en examinant les termes polynomiaux à intégrer. On suppose que l'élément physique a la même forme que l'élément de référence, soit un rectangle dans le cas du quadrangle (Fig. VIII.6). Avec cette hypothèse, le gradient de la transformation \mathbf{F} est constant ainsi que son déterminant. En effet, la transformation pour passer de l'élément de référence à l'élément physique s'écrit

$$x = \frac{l_x}{2}r; \quad y = \frac{l_y}{2}s$$

Le gradient de la transformation vaut alors

$$\mathbf{F} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial s} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial s} \end{bmatrix} = \begin{bmatrix} \frac{l_x}{2} & 0 \\ 0 & \frac{l_y}{2} \end{bmatrix}$$

Le déterminant vaut

$$\det \mathbf{F} = \frac{l_x l_y}{4}$$

Avec cette hypothèse, \mathbf{F} est constant et \mathbf{B}^e ne fait intervenir que les dérivées de la base canonique polynomiale du déplacement. En prenant de plus un matériau homogène dans l'élément, \mathbf{C} est aussi constant. Finalement, les termes de $\mathbf{B}^{eT}(r, s) \mathbf{C}^e \mathbf{B}^e(r, s) \det \mathbf{F}$ à intégrer ne font intervenir que les termes présents dans $\mathbf{B}^{eT}(r, s)$ multipliés un à un. Les plus haut degrés en r et s donnent le nombre de points de Gauss à prendre dans chaque direction.

Exercice VIII.5 Trouver les positions et les poids des points de Gauss pour l'élément quadrangle à 4 nœuds.

Solution :

La base canonique polynomiale d'approximation du déplacement du quadrangle à 4 nœuds est $[1 \ r \ s \ rs]$; en dérivant cette base par rapport à r d'une part et s d'autre part, on obtient les termes présents dans $\mathbf{B}(r, s)$ soient $[1 \ r \ s]$. Les termes à intégrer sont alors :

$$[1 \ r \ s] \times [1 \ r \ s] = [1 \ r \ s \ r^2 \ s^2 \ rs]$$

Les termes de plus haut degré dans chaque direction sont 2 pour r et 2 pour s dans ce cas. On trouve le nombre de points de Gauss n en écrivant $2n - 1 > 2$ soit $n = 2$ dans chaque direction.

L'expression du calcul de la matrice de rigidité est alors pour le quadrangle à 4 nœuds

$$\mathbf{K}^e = \sum_{i=1}^2 \sum_{j=1}^2 h w_i w_j \mathbf{B}^{eT}(r_i, s_j) \mathbf{C}^e \mathbf{B}^e(r_i, s_j) \det \mathbf{F}(r_i, s_j)$$

soit aussi

$$\mathbf{K}^e = \sum_{g=1}^4 h w_g \mathbf{B}^{eT}(r_g, s_g) \mathbf{C}^e \mathbf{B}^e(r_g, s_g) \det \mathbf{F}(r_g, s_g)$$

où les 4 points d'intégration sont :

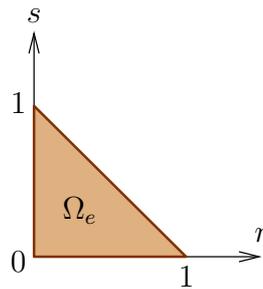
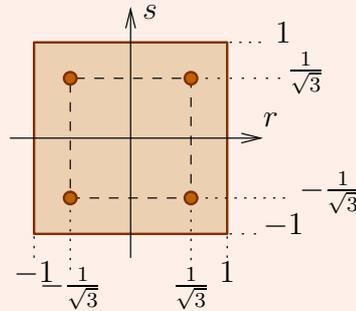


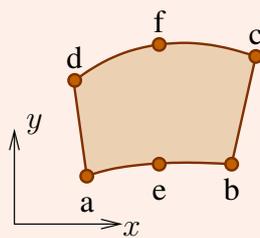
Figure VIII.7 – Éléments triangle de référence.

r	s	poids
0,5773502692	0,5773502692	1
0,5773502692	-0,5773502692	1
-0,5773502692	0,5773502692	1
-0,5773502692	-0,5773502692	1

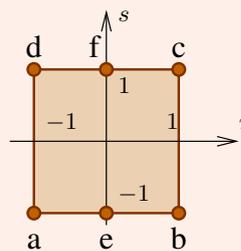


Exercice VIII.6 Trouver les positions et les poids des points de Gauss pour l'élément suivant dont la base canonique d'approximation est $[1 \ r \ r^2] \times [1 \ s]$

élément physique



élément de référence



VIII.3.2 Cas des éléments triangles

Dans le cas de l'élasticité plane, la matrice de rigidité élémentaire d'un triangle vaut :

$$\mathbf{K}^e = \int_{\Omega_{ref}} h^e \mathbf{B}^{eT}(r, s) \mathbf{C}^e \mathbf{B}^e(r, s) \det \mathbf{F} dr ds$$

où Ω_{ref} est le domaine triangulaire de référence (Fig. VIII.7) ; les variables r et s sont couplées. Les points d'intégration sont placés dans le triangle suivant des règles particulières. Le tableau VIII.2 donnent les positions et poids suivant le degré des polynômes à intégrer. Il existe

degré d'intégration	r	s	poids
1	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{2}$
2	$\frac{1}{2}$ 0 $\frac{1}{2}$	$\frac{1}{2}$ $\frac{1}{2}$ 0	$\frac{1}{6}$ $\frac{1}{6}$ $\frac{1}{6}$
3	$\frac{1}{3}$ 0,6 0,2 0,2	$\frac{1}{3}$ 0,2 0,6 0,2	$-\frac{27}{96}$ $\frac{25}{96}$ $\frac{25}{96}$ $\frac{25}{96}$

Tableau VIII.2 – Positions et poids des points d'intégration d'un triangle.

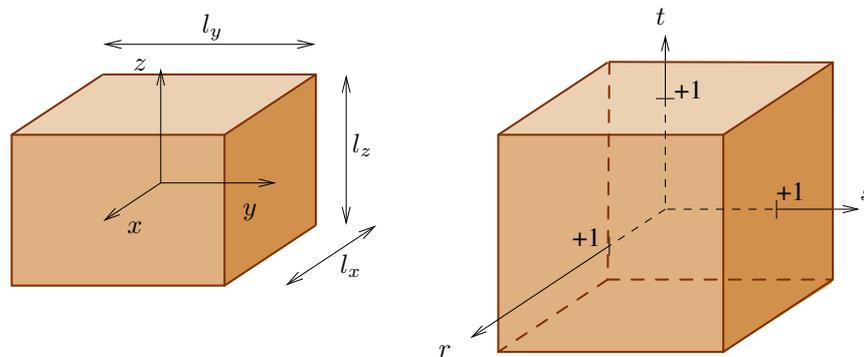


Figure VIII.8 – Élément hexaèdre physique de forme parallélépipède rectangle et élément de référence hexaèdre.

d'autres règles développées par différents auteurs et pour des degrés supérieurs, on ne donne ici que quelques exemples de positions et poids.

La somme des poids donne toujours l'aire du triangle de référence, soit 0,5.

VIII.4 Intégration des fonctions à trois variables

VIII.4.1 Cas des éléments hexaèdres

Dans le cas de l'élasticité tridimensionnelle, la matrice de rigidité élémentaire d'un hexaèdre vaut :

$$\mathbf{K}^e = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \mathbf{B}^{eT}(r, s, t) \mathbf{C}^e \mathbf{B}^e(r, s, t) \det \mathbf{F} dr ds dt .$$

Le choix du nombre de points de Gauss se fait en examinant les termes polynomiaux à intégrer. On suppose que l'élément physique a la même forme que l'élément de référence, soit un parallélépipède rectangle dans ce cas (Fig. VIII.8).

Avec cette hypothèse, \mathbf{F} est constant et \mathbf{B}^e ne fait intervenir que les dérivées de la base canonique polynomiale du déplacement. En effet la transformation pour passer de l'élément de

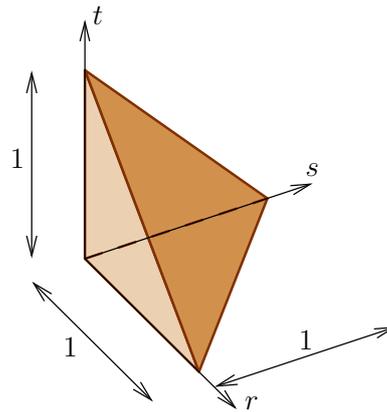


Figure VIII.9 – Élément tétraèdre de référence.

référence à l'élément physique s'écrit

$$x = \frac{l_x}{2}r; \quad y = \frac{l_y}{2}s; \quad z = \frac{l_z}{2}t$$

Le gradient de la transformation vaut alors

$$\mathbf{F} = \begin{bmatrix} \frac{l_x}{2} & 0 & 0 \\ 0 & \frac{l_y}{2} & 0 \\ 0 & 0 & \frac{l_z}{2} \end{bmatrix}$$

Le déterminant vaut

$$\det \mathbf{F} = \frac{l_x l_y l_z}{8}$$

En prenant de plus un matériau homogène dans l'élément, \mathbf{C} est aussi constant. Finalement, les termes de $\mathbf{B}^{eT}(r, s, t) \mathbf{C}^e \mathbf{B}^e(r, s, t) \det \mathbf{F}$ à intégrer ne font intervenir que les termes présents dans $\mathbf{B}^{eT}(r, s, t)$ multipliés un à un. Les plus haut degrés en r , s et t donnent le nombre de points de Gauss à prendre dans chaque direction.

VIII.4.2 Cas des éléments tétraèdres

Dans le cas de l'élasticité tridimensionnelle, la matrice de rigidité élémentaire d'un tétraèdre vaut :

$$\mathbf{K}^e = \int_{\Omega_{ref}} h^e \mathbf{B}^{eT}(r, s, t) \mathbf{C}^e \mathbf{B}^e(r, s, t) \det \mathbf{F} \, dr \, ds \, dt$$

où Ω_{ref} est le domaine tétraédrique de référence (Fig. VIII.9); les variables r , s et t sont couplées. Les points d'intégrations sont placés dans le tétraèdre suivant des règles particulières. Le tableau VIII.3 donnent les positions et poids suivant le degré des polynômes à intégrer. Il existe d'autres règles développées par différents auteurs et pour des degrés supérieurs, on ne donne ici que quelques exemples de positions et poids.

La somme des poids donne toujours le volume du tétraèdre de référence, soit $\frac{1}{6}$.

VIII.5 Critères de qualité du maillage

Dans la pratique, les éléments ne sont jamais de la même forme que l'élément de référence, donc le gradient de la transformation \mathbf{F} n'est pas constant dans l'élément et dépend des coordonnées

degré d'intégration	r	s	t	pois
1	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{6}$
2	0,58541020 0,138119660 0,138119660 0,138119660	0,138119660 0,58541020 0,138119660 0,138119660	0,138119660 0,138119660 0,58541020 0,138119660	$\frac{1}{24}$ $\frac{1}{24}$ $\frac{1}{24}$ $\frac{1}{24}$

Tableau VIII.3 – Positions et poids des points d'intégration d'un tétraèdre.

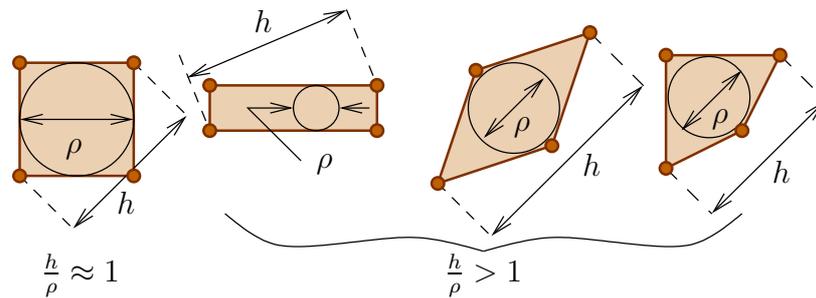


Figure VIII.10 – Taux de distorsion pour le quadrangle.

de référence (r, s, t) . L'opérateur gradient discrétisé, qui est calculé à partir de \mathbf{F}^{-1} , contient alors des fractions polynomiales en (r, s, t) . Finalement, le terme " $\mathbf{B}^T \mathbf{C} \mathbf{B} \det \mathbf{F}$ " contient aussi des fractions polynomiales en (r, s, t) , qui ne peut pas s'intégrer exactement par points d'intégration. Cela engendre des erreurs dans le calcul numérique par points d'intégration de la matrice de rigidité. Cette erreur est maîtrisée en contrôlant la qualité du maillage : les éléments doivent être le plus proche possible de la forme de l'élément de référence associé.

Plusieurs critères sont utilisés pour garantir la qualité du maillage. Par exemple pour le quadrangle, on peut utiliser le taux de distorsion d'un élément (Fig. VIII.10) défini comme le rapport entre la plus grande longueur h sur le diamètre du plus grand cercle inscrit ρ ; le critère est alors :

$$\frac{h}{\rho} < \text{valeur donnée (de 3 à 10 généralement)}$$

Un autre critère utilisé concerne le respect des angles dans l'élément quadrangle : les angles de l'élément physique doivent être les plus proches possibles d'angles droits.

D'autres critères de qualité de maillage propres à chaque logiciel de maillage et à chaque type d'élément existent, ils sont détaillés dans les notices d'utilisation.

IX — Analyse des résultats E.F.

IX.1 Post-traitement, lissage des contraintes

IX.1.1 Propriétés de la solution éléments finis

On rappelle que le champ de déplacement solution $\vec{u}^h \in \mathcal{V}^h$ du problème discrétisé est cinématiquement admissible ($\vec{u}^h \in \mathcal{U}^h$) et minimise l'erreur en énergie avec la solution du problème de référence. Le champ de contrainte correspondant $\underline{\sigma}^h = \underline{C}\underline{\epsilon}(\vec{u}^h)$ n'est par contre pas statiquement admissible et est de plus discontinu d'un élément à l'autre. Afin de rendre ce champ plus proche de la solution exacte, on peut le rendre continu par "lissage". Ce nouveau champ de contrainte, noté $\tilde{\underline{\sigma}}$, n'est pas statiquement admissible, mais appartient à l'ensemble \mathcal{C} des champs de contraintes définis précédemment (Fig. IX.1).

IX.1.2 Contrainte lissée

Les contraintes obtenues à partir du champ de déplacement \vec{u}^h solution du problème éléments finis s'écrivent :

$$\underline{\sigma}^h = \underline{C}\underline{\epsilon}(\vec{u}^h) = \underline{C}\underline{B}\underline{U}$$

où \underline{C} est la matrice de comportement et \underline{B} est l'opérateur gradient discrétisé.

On cherche une approximation des contraintes $\tilde{\underline{\sigma}}$ basée sur les fonctions de forme de \mathcal{V}^h . On note $\tilde{\sigma}$ une des composante de $\tilde{\underline{\sigma}}$ et σ^h une des composantes de $\underline{\sigma}^h$. Autrement dit, on cherche $\tilde{\sigma}$ dans un espace \mathcal{C}^h défini par

$$\mathcal{C}^h = \text{Vect}\{\varphi_1, \varphi_2, \dots, \varphi_n\}$$

où n est le nombre de nœuds du maillage.

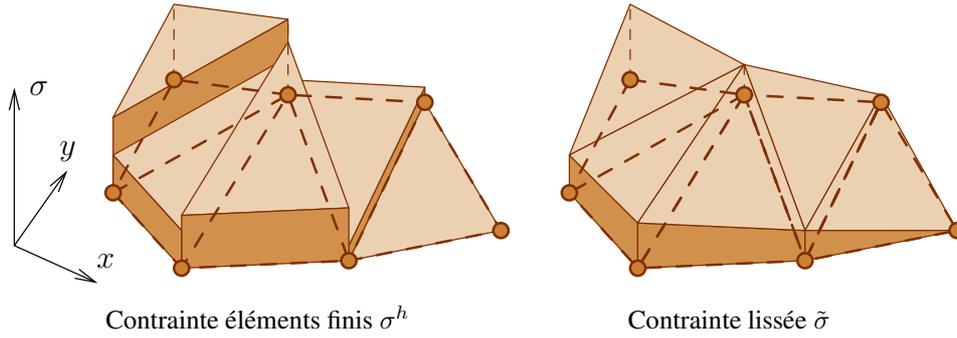


Figure IX.1 – Contrainte obtenue par éléments finis et contrainte lissée pour les triangles à 3 nœuds.

La contrainte lissée $\tilde{\sigma}$ s'écrit alors sous la forme :

$$\tilde{\sigma} = [\varphi_1 \ \varphi_2 \ \dots \ \varphi_n] \begin{bmatrix} \tilde{\sigma}_1 \\ \tilde{\sigma}_2 \\ \vdots \\ \tilde{\sigma}_n \end{bmatrix} = \varphi \underline{\Sigma} = \underline{\Sigma}^T \varphi^T$$

où n est le nombre de nœuds du maillage et où $\underline{\Sigma}$ contient les contraintes aux nœuds du maillage ($\underline{\Sigma}$ est la représentation de $\tilde{\sigma}$ dans la base des fonctions de forme).

L'objectif est de trouver $\tilde{\sigma}$ (ou de façon équivalente $\underline{\Sigma}$) permettant d'approcher au mieux σ^h . Ceci est réalisé par une minimisation par moindres carrés de la fonction "distance" entre $\tilde{\sigma}$ et σ^h définie par la fonction J suivante :

$$J(\tilde{\sigma}) = \frac{1}{2} \int_{\Omega} (\tilde{\sigma} - \sigma^h)^2 d\Omega$$

soit aussi

$$J(\tilde{\sigma}) = \frac{1}{2} \int_{\Omega} (\tilde{\sigma} - \sigma^h)(\tilde{\sigma} - \sigma^h) d\Omega$$

ou encore

$$J(\underline{\Sigma}) = \frac{1}{2} \int_{\Omega} (\underline{\Sigma}^T \varphi^T - \sigma^h)(\varphi \underline{\Sigma} - \sigma^h) d\Omega$$

$$J(\underline{\Sigma}) = \frac{1}{2} \underline{\Sigma}^T \underbrace{\int_{\Omega} \varphi^T \varphi d\Omega}_{\mathbf{A}} \underline{\Sigma} - \underline{\Sigma}^T \underbrace{\int_{\Omega} \varphi^T \sigma^h d\Omega}_{\mathbf{b}} + \frac{1}{2} \underbrace{\int_{\Omega} \sigma^{h2} d\Omega}_{c}$$

$$J(\underline{\Sigma}) = \frac{1}{2} \underline{\Sigma}^T \mathbf{A} \underline{\Sigma} - \underline{\Sigma}^T \mathbf{b} + c$$

En utilisant la notation indicielle on a

$$J(\underline{\Sigma}) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \Sigma_i A_{ij} \Sigma_j - \sum_{i=1}^n \Sigma_i b_i + c$$

La solution $\underline{\Sigma}$ minimise $J(\underline{\Sigma})$, ce qui revient à trouver $\underline{\Sigma}$ tel que

$$\forall \alpha \in [1, \dots, n], \quad \frac{\partial J(\underline{\Sigma})}{\partial \Sigma_{\alpha}} = 0$$

C'est à dire

$$\frac{\partial J(\underline{\Sigma})}{\partial \Sigma_{\alpha}} = \frac{1}{2} \sum_{j=1}^n A_{\alpha j} \Sigma_j + \frac{1}{2} \sum_{i=1}^n \Sigma_i A_{i\alpha} - b_{\alpha} = 0$$

La matrice \mathbf{A} étant symétrique ($A_{ij} = A_{ji}$), on a alors :

$$\frac{\partial J(\underline{\Sigma})}{\partial \Sigma_\alpha} = \sum_{i=1}^n A_{\alpha i} \Sigma_i - b_\alpha = 0$$

En faisant varier α de 1 à n , on obtient un système de n équations à n inconnues que l'on peut écrire sous la forme matricielle suivante :

$$\mathbf{A}\underline{\Sigma} = \underline{b}$$

IX.1.3 Implémentation et résolution

La matrice \mathbf{A} et le vecteur \underline{b} sont calculés numériquement en utilisant toute la stratégie de calcul éléments finis (élément de référence, points d'intégration, assemblage ...).

Dans la pratique, la matrice \mathbf{A} est rendue diagonale en plaçant sur la diagonale la somme des termes de la colonne correspondante.

Dans l'élément e , la matrice \mathbf{A}_e vaut

$$\mathbf{A}_e = \int_{\Omega_e} \begin{bmatrix} \varphi_a \\ \varphi_b \\ \vdots \\ \varphi_{n_e} \end{bmatrix} \begin{bmatrix} \varphi_a & \varphi_b & \cdots & \varphi_{n_e} \end{bmatrix} d\Omega_e$$

où n_e est le nombre de nœuds connectés à l'élément e . Soit aussi

$$\mathbf{A}_e = \int_{\Omega_e} \begin{bmatrix} \varphi_a^2 & \varphi_a \varphi_b & \cdots & \varphi_a \varphi_{n_e} \\ \varphi_a \varphi_b & \varphi_b^2 & \cdots & \varphi_b \varphi_{n_e} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{n_e} \varphi_a & \varphi_{n_e} \varphi_b & \cdots & \varphi_{n_e}^2 \end{bmatrix} d\Omega_e$$

En sommant les colonnes sur la diagonale on a

$$\mathbf{A}_e = \int_{\Omega_e} \begin{bmatrix} \varphi_a(\varphi_a + \varphi_b + \cdots + \varphi_{n_e}) & 0 & \cdots & 0 \\ 0 & \varphi_b(\varphi_a + \varphi_b + \cdots + \varphi_{n_e}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \varphi_{n_e}(\varphi_a + \varphi_b + \cdots + \varphi_{n_e}) \end{bmatrix} d\Omega_e$$

Or la somme des fonctions de base fait toujours 1, la matrice élémentaire \mathbf{A}_e devient alors

$$\mathbf{A}_e = \int_{\Omega_e} \begin{bmatrix} \varphi_a & 0 & \cdots & 0 \\ 0 & \varphi_b & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \varphi_{n_e} \end{bmatrix} d\Omega_e$$

Après assemblage, on obtient la matrice \mathbf{A} "diagonalisée" par sommation des colonnes sur la diagonale suivante :

$$\mathbf{A} = \int_{\Omega} \begin{bmatrix} \varphi_1 & 0 & \cdots & 0 \\ 0 & \varphi_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \varphi_n \end{bmatrix} d\Omega$$

Dans l'élément e , le second membre \underline{b}_e du système vaut

$$\underline{b}_e = \int_{\Omega_e} \begin{bmatrix} \varphi_a \\ \varphi_b \\ \vdots \\ \varphi_{n_e} \end{bmatrix} \sigma^h d\Omega_e$$

et doit être calculé pour chaque composante de l'opérateur des contraintes. Après assemblage, le second membre devient

$$\underline{b} = \int_{\Omega} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_n \end{bmatrix} \sigma^h d\Omega$$

Le système d'équations étant rendu diagonal, la résolution est immédiate et donne au nœud i et pour chaque composante de l'opérateur des contraintes :

$$\tilde{\sigma}_i = \frac{\int_{\Omega} \varphi_i \sigma^h d\Omega}{\int_{\Omega} \varphi_i d\Omega}.$$

IX.1.4 Application au triangle à 3 nœuds

Afin d'illustrer la démarche, on se place dans le cadre de l'élasticité plane avec une discrétisation par des triangles à 3 nœuds. Les contraintes obtenues à partir du champ de déplacement \vec{u}^h solution du problème éléments finis s'écrivent :

$$\underline{\sigma}^h = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} = \underline{C} \underline{\epsilon}(\vec{u}^h) = \underline{C} \underline{B} \underline{U}$$

où \underline{C} est la matrice de comportement et \underline{B} est l'opérateur gradient discrétisé. Pour le triangle à 3 nœuds, l'opérateur \underline{B} étant constant dans un élément, les contraintes sont constantes dans un élément.

La matrice élémentaire \underline{A}_e vaut

$$\underline{A}_e = \int_{\Omega_e} \begin{bmatrix} \varphi_a & 0 & 0 \\ 0 & \varphi_b & 0 \\ 0 & 0 & \varphi_c \end{bmatrix} d\Omega_e = \frac{a_e}{3} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

où a_e est l'aire de l'élément triangle.

Dans l'élément e , le second membre \underline{b}_e du système pour la contrainte σ_{xx} vaut

$$\underline{b}_e^{xx} = \int_{\Omega_e} \begin{bmatrix} \varphi_a \\ \varphi_b \\ \varphi_c \end{bmatrix} \sigma_{(e)xx}^h d\Omega_e = \sigma_{(e)xx}^h \frac{a_e}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

où $\sigma_{(e)xx}^h$ est la valeur de la contrainte calculée à partir de la solution en déplacements \vec{u}^h dans l'élément e . De même, on a pour les contraintes σ_{yy} et σ_{xy} :

$$\underline{b}_e^{yy} = \sigma_{(e)yy}^h \frac{a_e}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{et} \quad \underline{b}_e^{xy} = \sigma_{(e)xy}^h \frac{a_e}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

En notant \mathcal{E}_i l'ensemble des éléments connectés au nœud i , on obtient que les contraintes lissées au nœud i du maillage valent :

$$\begin{aligned} \tilde{\sigma}_{(i)xx} &= \frac{\sum_{e \in \mathcal{E}_i} a_e \sigma_{(e)xx}^h}{\sum_{e \in \mathcal{E}_i} a_e}, \\ \tilde{\sigma}_{(i)yy} &= \frac{\sum_{e \in \mathcal{E}_i} a_e \sigma_{(e)yy}^h}{\sum_{e \in \mathcal{E}_i} a_e}, \\ \tilde{\sigma}_{(i)xy} &= \frac{\sum_{e \in \mathcal{E}_i} a_e \sigma_{(e)xy}^h}{\sum_{e \in \mathcal{E}_i} a_e}. \end{aligned}$$

IX.2 Estimateur d'erreur

IX.2.1 Rappel sur l'erreur en déplacement

L'erreur en déplacement \vec{e} est définie comme la différence entre la solution de référence $\vec{u}^{ex} \in \mathcal{V}$ et la solution éléments finis $\vec{u}^h \in \mathcal{V}^h$:

$$\vec{e} = \vec{u}^{ex} - \vec{u}^h$$

L'erreur en énergie e est donnée par la norme énergétique de \vec{e} :

$$e = \|\vec{e}\|_E$$

On définit l'erreur relative globale ξ comme

$$\xi = \frac{\|\vec{e}\|_E}{\|\vec{u}^{ex}\|_E}$$

donc

$$\xi^2 = \frac{\|\vec{e}\|_E^2}{\|\vec{u}^{ex}\|_E^2} = \frac{\|\vec{u}^{ex} - \vec{u}^h\|_E^2}{\|\vec{u}^{ex}\|_E^2}.$$

IX.2.2 Estimateur à priori

On peut montrer que l'erreur relative globale est de l'ordre de h^p :

$$\xi \approx c h^p$$

où c , h et p sont définis ainsi :

- c est une constante qui dépend du problème ;
- h est la longueur relative des éléments : c'est le rapport de la longueur des éléments sur la longueur caractéristique du domaine discrétisé, donc $h < 1$;

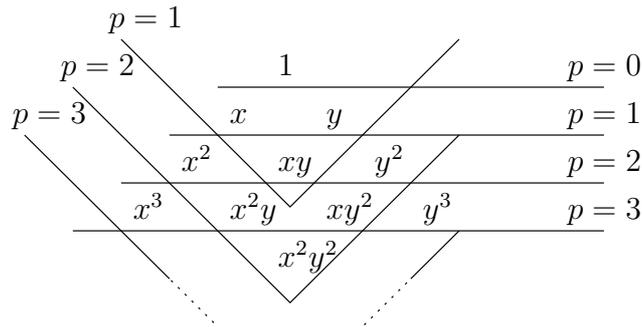


Figure IX.2 – Complétude de la base d'approximation pour les polynômes à 2 variables.

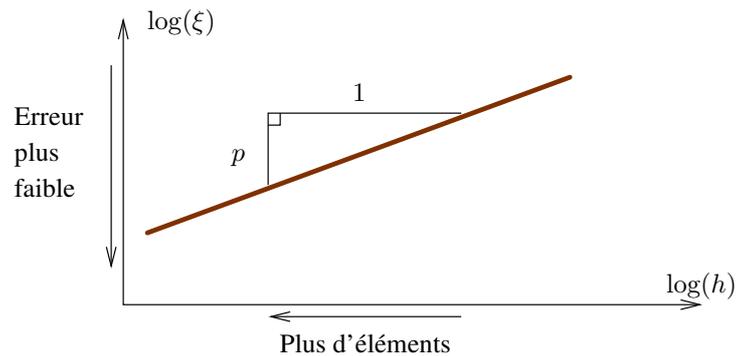


Figure IX.3 – Diagramme log – log de l'erreur relative en fonction de h .

- p est l'ordre maximal de la base polynomiale complète de l'espace d'approximation \mathcal{V}^h : pour les polynômes à 2 variables, p est tel que tous les termes $x^\alpha y^\beta$ avec $\alpha + \beta = p$ soient présents dans la base d'approximation du déplacement (Fig. IX.2).

Deux stratégies sont possibles afin de diminuer l'erreur ξ :

- diminuer h , c'est à dire augmenter le nombre d'éléments, on parle de *méthode h* ou *h -method* : on augmente la taille du problème sans changer de type d'élément,
- augmenter p , c'est à dire augmenter l'ordre d'approximation de la base, on parle de *méthode p* ou *p -method* : on change de type d'élément et on augmente indirectement la taille du problème puisque les éléments possèdent alors plus de nœuds.

De façon graphique, l'erreur ξ peut être représentée en fonction de h dans un diagramme log – log, en effet :

$$\xi \approx ch^p \Rightarrow \log(\xi) \approx \log(c) + p \log(h)$$

Dans ce diagramme, l'erreur suit une droite de pente p (Fig. IX.3). Lorsque p augmente, la droite devient plus pentue et la convergence vers la solution exacte est plus rapide. Le taux de convergence caractérise la pente de la droite $\log(c) + p \log(h)$, il vaut donc p . Par exemple pour un triangle à 6 nœuds ($p = 2$), l'erreur en énergie est divisée par 100 si l'on multiplie par 10 le nombre d'éléments.

IX.2.3 Estimateur à posteriori

Dans le cas général, la solution exacte n'est pas connue. On peut cependant calculer une estimation de l'erreur. Plusieurs estimateurs existent, on présente ici un estimateur simple à mettre en œuvre

appelé *ZZI*. Cet estimateur est fréquemment utilisée par les codes de calculs. Il a par contre l'inconvénient de ne pas assurer de donner une estimation de l'erreur à coup sûr supérieure à l'erreur vraie, autrement dit, il peut sous-estimer l'erreur vraie.

On rappelle que l'erreur relative globale ξ vaut

$$\xi^2 = \frac{\| \underline{u}^{ex} - \underline{u}^h \|^2}{\| \underline{u}^{ex} \|^2} = \frac{a(\underline{u}^{ex} - \underline{u}^h, \underline{u}^{ex} - \underline{u}^h)}{a(\underline{u}^{ex}, \underline{u}^{ex})}$$

Le dénominateur vaut

$$a(\underline{u}^{ex}, \underline{u}^{ex}) = \int_{\Omega} \underline{\epsilon}^T(\underline{u}^{ex}) \mathbf{C} \underline{\epsilon}(\underline{u}^{ex}) \, d\Omega$$

La relation entre la contrainte exacte et le champ de déformations du déplacement exact \underline{u}^{ex} est :

$$\underline{\sigma}^{ex} = \mathbf{C} \underline{\epsilon}(\underline{u}^{ex})$$

ou encore

$$\underline{\epsilon}(\underline{u}^{ex}) = \mathbf{C}^{-1} \underline{\sigma}^{ex}$$

Donc en remplaçant dans l'expression de $a(\underline{u}^{ex}, \underline{u}^{ex})$, on a :

$$a(\underline{u}^{ex}, \underline{u}^{ex}) = \int_{\Omega} \underline{\sigma}^{exT} \mathbf{C}^{-1} \underline{\sigma}^{ex} \, d\Omega$$

Le numérateur vaut

$$a(\underline{u}^{ex} - \underline{u}^h, \underline{u}^{ex} - \underline{u}^h) = \int_{\Omega} \underline{\epsilon}^T(\underline{u}^{ex} - \underline{u}^h) \mathbf{C} \underline{\epsilon}(\underline{u}^{ex} - \underline{u}^h) \, d\Omega$$

soit aussi

$$a(\underline{u}^{ex} - \underline{u}^h, \underline{u}^{ex} - \underline{u}^h) = \int_{\Omega} \left(\underline{\epsilon}^T(\underline{u}^{ex}) - \underline{\epsilon}^T(\underline{u}^h) \right) \mathbf{C} \left(\underline{\epsilon}(\underline{u}^{ex}) - \underline{\epsilon}(\underline{u}^h) \right) \, d\Omega$$

En suivant la même démarche que pour le dénominateur, on a :

$$a(\underline{u}^{ex} - \underline{u}^h, \underline{u}^{ex} - \underline{u}^h) = \int_{\Omega} \left(\underline{\sigma}^{exT} - \underline{\sigma}^hT \right) \mathbf{C}^{-1} \left(\underline{\sigma}^{ex} - \underline{\sigma}^h \right) \, d\Omega$$

Finalement, l'erreur relative globale ξ vaut

$$\xi^2 = \frac{\int_{\Omega} \left(\underline{\sigma}^{exT} - \underline{\sigma}^hT \right) \mathbf{C}^{-1} \left(\underline{\sigma}^{ex} - \underline{\sigma}^h \right) \, d\Omega}{\int_{\Omega} \underline{\sigma}^{exT} \mathbf{C}^{-1} \underline{\sigma}^{ex} \, d\Omega}$$

La contrainte exacte $\underline{\sigma}^{ex}$ n'est pas connue, mais l'idée est de la remplacer par la contrainte lissée $\tilde{\underline{\sigma}}$. La contrainte lissée est en effet plus régulière que la contrainte $\underline{\sigma}(\underline{u}^h)$, elle approche mieux la réalité que $\underline{\sigma}(\underline{u}^h)$.

En remplaçant $\underline{\sigma}^{ex}$ par $\tilde{\underline{\sigma}}$ on obtient une estimation de l'erreur relative ξ , on note cette estimation θ :

$$\theta^2 = \frac{\int_{\Omega} \left(\tilde{\underline{\sigma}}^T - \underline{\sigma}^hT \right) \mathbf{C}^{-1} \left(\tilde{\underline{\sigma}} - \underline{\sigma}^h \right) \, d\Omega}{\int_{\Omega} \tilde{\underline{\sigma}}^T \mathbf{C}^{-1} \tilde{\underline{\sigma}} \, d\Omega}$$

En notant θ_e la quantité définie pour l'élément e telle que

$$\theta_e^2 = \frac{\int_{\Omega_e} \left(\tilde{\underline{\sigma}}^T - \underline{\sigma}^hT \right) \mathbf{C}^{-1} \left(\tilde{\underline{\sigma}} - \underline{\sigma}^h \right) \, d\Omega}{\int_{\Omega_e} \tilde{\underline{\sigma}}^T \mathbf{C}^{-1} \tilde{\underline{\sigma}} \, d\Omega}$$

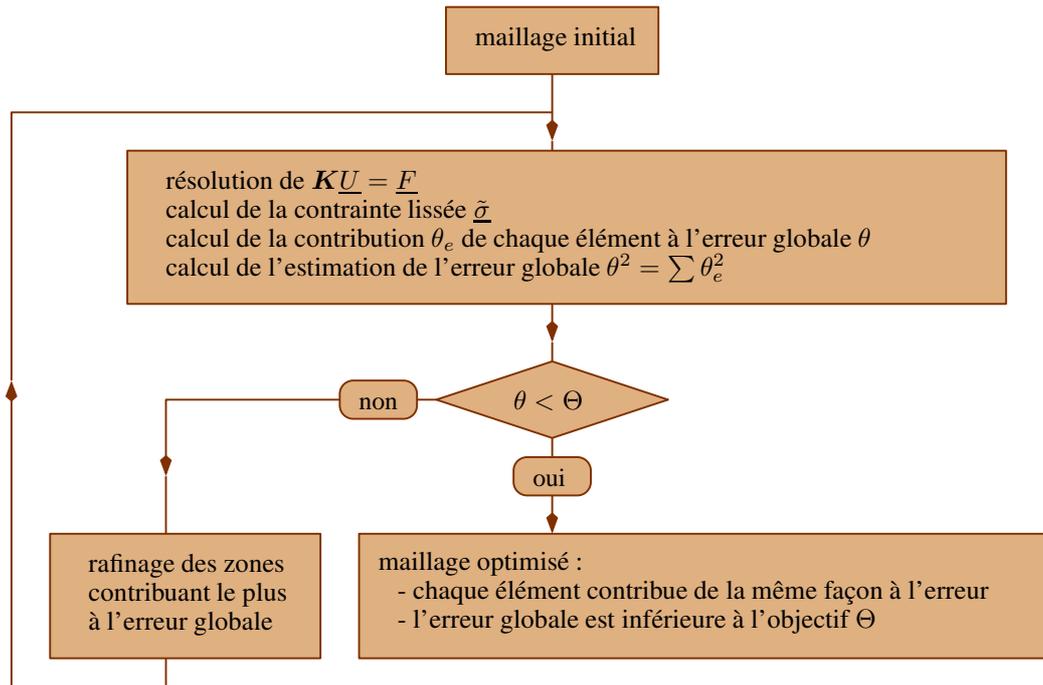


Figure IX.4 – Processus itératif de maillage adaptatif.

où Ω_e est le domaine de l'élément e , on a

$$\theta^2 = \sum_{e=1}^{\text{nb. elts}} \theta_e^2$$

La quantité θ_e est la contribution de l'élément e à l'erreur relative globale.

IX.2.4 Maillage adaptatif

On peut faire des itérations sur la construction du maillage à partir du résultat d'un calcul, notamment en calculant l'estimateur d'erreur globale ainsi que les contributions θ_e des éléments à l'erreur relative globale θ . Le maillage est alors raffiné dans les zones qui contribuent le plus à l'erreur globale. La stratégie de maillage adaptatif peut alors être automatisée afin que chaque élément contribue de la même façon à l'erreur globale tout en ayant une erreur globale relative inférieure à une valeur objectif fixée par l'utilisateur et notée Θ . Ce processus itératif est détaillé sur la figure IX.4.

X — Éléments finis de barre et poutre

X.1 Modèle de poutre

X.1.1 Modèle géométrique

La figure X.1 montre un assemblage de poutres permettant de construire une charpente métallique. Une poutre est un solide dont une dimension est plus grande que les 2 autres.



Figure X.1 – Charpente constituée d'un assemblage de poutres

Dans le cadre de ce cours, on ne s'intéresse qu'aux poutres droites. De plus, on suppose que les poutres ont des sections constantes.

On étudie un élément fini de poutre connecté à 2 nœuds a et b (Fig. X.2). Un repère local $(\vec{x}, \vec{y}, \vec{z})$ est attaché à l'élément, l'axe \vec{x} coïncide avec la ligne moyenne de la poutre. L'élément a pour longueur L suivant \vec{x} et a pour section constante S . Le domaine tridimensionnel Ω associé à la poutre est alors

$$\Omega = [0, L] \times S$$

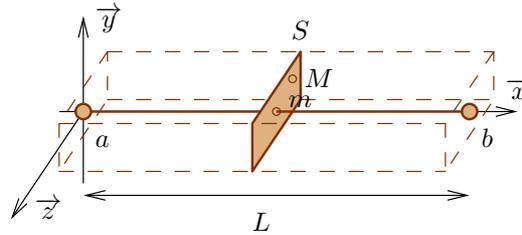


Figure X.2 – Élément de poutre

X.1.2 Hypothèses cinématiques

On note $M(x, y, z)$ un point de Ω et $m(x)$ sa projection sur sa ligne moyenne. On se restreint dans ce cours à des mouvements dans le plan (x, y) d'une section S de la poutre. On fait l'hypothèse qu'une section S reste plane après déformation, et qu'elle a un mouvement de solide rigide :

- translation suivant \vec{x} (notée $u(x)$) et translation suivant \vec{y} (notée $v(x)$),
- rotation autour de \vec{z} d'un angle $\theta = -\beta$ qui ne dépend que de x .

On note $\vec{u}_0(m) = u(x)\vec{x} + v(x)\vec{y}$ le déplacement de m . En supposant que les déplacements et les rotations sont petits, le déplacement de M s'écrit :

$$\vec{u}(M) = \vec{u}_0(m) + \overline{Mm} \wedge \vec{\Omega} = u(x)\vec{x} + v(x)\vec{y} - (y\vec{y} + z\vec{z}) \wedge (-\beta(x)\vec{z})$$

donc

$$\vec{u}(M) = u(x)\vec{x} + v(x)\vec{y} + y\beta(x)\vec{x}$$

soit

$$\vec{u}(M) = \begin{bmatrix} u(x) + y\beta(x) \\ v(x) \end{bmatrix}_{(\vec{x}, \vec{y})}$$

On pose alors l'ensemble \mathcal{V} des champs de fonctions "suffisamment régulières" définis de $[0, L]$ vers \mathbb{R} :

$$\mathcal{V} = \left\{ \begin{array}{l} f \text{ "réguliers"} : [0, L] \rightarrow \mathbb{R} \\ x \mapsto f(x) \end{array} \right\}$$

Le champ de déplacement de la poutre est alors $\vec{v}(M)$ tel que

$$\forall x \in [0, L] ; \forall (y, z) \in S,$$

$$\vec{v}(x, y, z) = (u(x) + y\beta(x))\vec{x} + v(x)\vec{y}$$

avec $u, v, \beta \in \mathcal{V}$.

X.1.3 Écriture de la relation déformations-déplacement

Les déformations associées à un champ de déplacement \vec{u} de la poutre sont calculées à partir de :

$$\epsilon(\vec{u}) = \frac{1}{2}(\mathbf{Grad}(\vec{u}) + \mathbf{Grad}^T(\vec{u}))$$

soient :

$$\epsilon_{xx} = \frac{\partial}{\partial x}(u + y\beta) = u_{,x} + y\beta_{,x}$$

$$2\epsilon_{xy} = \frac{\partial}{\partial y}(u + y\beta) + \frac{\partial}{\partial x}v = \beta + v_{,x}$$

$$\epsilon_{yy} = \epsilon_{zz} = \epsilon_{xz} = \epsilon_{yz} = 0.$$

Seules les déformations ϵ_{xx} et ϵ_{xy} ne sont pas nulles. On note alors les déformations sous la forme

$$\underline{\epsilon} = \begin{bmatrix} \epsilon_{xx} \\ 2\epsilon_{xy} \end{bmatrix}.$$

En notant

- la déformation de membrane $e = u_{,x}$,
- la courbure $\chi = \beta_{,x}$,
- et la déformation transverse $\gamma = v_{,x} + \beta$,

on a les relations suivantes reliant ϵ_{xx} et $2\epsilon_{xy}$ avec e , χ et γ :

$$\epsilon_{xx} = e + y\chi, \quad \text{et} \quad 2\epsilon_{xy} = \gamma.$$

X.1.4 Relation de comportement

Une poutre étant un milieu mince dans les deux directions \vec{y} et \vec{z} , et étant sollicitée seulement dans le plan (x, y) on suppose que les seules contraintes non nulles sont la contrainte axiale σ_{xx} et la contrainte de cisaillement transverse σ_{xy} . On note les contraintes sous la forme

$$\underline{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{xy} \end{bmatrix}.$$

La loi de Hooke

$$\underline{\epsilon} = \frac{1+\nu}{E} \underline{\sigma} - \frac{\nu}{E} \text{Tr}(\underline{\sigma}) \mathbf{1}$$

donne alors :

$$\epsilon_{xx} = \frac{1+\nu}{E} \sigma_{xx} - \frac{\nu}{E} \sigma_{xx} = \frac{1}{E} \sigma_{xx} \quad \text{et} \quad \epsilon_{xy} = \frac{1+\nu}{E} \sigma_{xy}$$

soit

$$\mathbf{C} = \begin{bmatrix} E & 0 \\ 0 & \frac{E}{2(1+\nu)} \end{bmatrix} = \begin{bmatrix} E & 0 \\ 0 & \mu \end{bmatrix}$$

X.1.5 Énergie de déformation

Avec les hypothèses précédentes, l'énergie de déformation vaut :

$$\mathcal{E}_D = \frac{1}{2} \int_{\Omega} \underline{\sigma}^T \underline{\epsilon} \, d\Omega = \frac{1}{2} \int_0^L \int_S [(e + y\chi) \quad \gamma] \mathbf{C} \begin{bmatrix} e + y\chi \\ \gamma \end{bmatrix} \, dS \, dx$$

$$\mathcal{E}_D = \frac{1}{2} \int_0^L \int_S (e + y\chi) E (e + y\chi) \, dS \, dx + \frac{1}{2} \int_0^L \int_S \gamma \mu \gamma \, dS \, dx$$

soit aussi en développant les expressions

$$\mathcal{E}_D = \frac{1}{2} \int_0^L \int_S E e^2 \, dS \, dx + \frac{1}{2} \int_0^L \int_S y^2 E \chi^2 \, dS \, dx + \frac{1}{2} \int_0^L \int_S 2y E e \chi \, dS \, dx + \frac{1}{2} \int_0^L \int_S \mu \gamma^2 \, dS \, dx$$

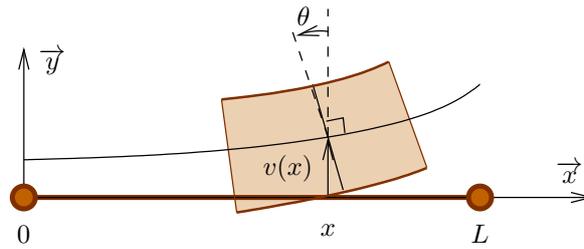


Figure X.3 – Hypothèse d'Euler-Bernoulli pour une poutre en flexion.

En remarquant que E , e , χ , μ et γ ne varient pas dans S , on a

$$\mathcal{E}_D = \frac{1}{2} \int_0^L E e^2 \int_S dS dx + \frac{1}{2} \int_0^L E \chi^2 \int_S y^2 dS dx + \frac{1}{2} \int_0^L E e \chi \int_S 2y dS dx + \frac{1}{2} \int_0^L \mu \gamma^2 \int_S dS dx$$

On remarque que :

$$\int_S dS = S, \text{ aire de la section ;}$$

$$\int_S y dS = 0, \text{ par définition du centre de section ;}$$

$$\int_S y^2 dS = I, \text{ moment quadratique de la section autour de } \vec{z}.$$

En remplaçant dans l'expression de l'énergie de déformation, on a alors :

$$\mathcal{E}_D = \mathcal{E}_D^m + \mathcal{E}_D^f + \mathcal{E}_D^{ct}$$

avec

$$\mathcal{E}_D^m = \frac{1}{2} \int_0^L ES e^2 dx ; \quad \mathcal{E}_D^f = \frac{1}{2} \int_0^L EI \chi^2 dx ; \quad \mathcal{E}_D^{ct} = \frac{1}{2} \int_0^L \mu S \gamma^2 dx$$

L'énergie de déformation peut s'écrire sous forme de trois contributions :

- \mathcal{E}_D^m : énergie de déformation de traction-compression (membrane),
- \mathcal{E}_D^f : énergie de déformation de flexion,
- \mathcal{E}_D^{ct} : énergie de déformation de cisaillement transverse.

X.1.6 Hypothèses sur les déformations transverses

Hypothèse d'Euler-Bernoulli

L'hypothèse d'Euler-Bernoulli suppose que la déformation transverse ϵ_{xy} est nulle. Ce modèle est valable pour les poutres dites "minces". On note $\theta = -\beta$. La relation qui relie θ et v pour que $\epsilon_{xy} = 0$ est :

$$\gamma = 2\epsilon_{xy} = \beta + v_{,x} = 0$$

ce qui entraîne que $\beta = -v_{,x}$ et donc que $\theta = v_{,x}$. Autrement dit, une section initialement perpendiculaire à la ligne moyenne reste perpendiculaire à la ligne moyenne déformée (Fig. X.3).

L'énergie de déformation en fonction de u et v vaut :

$$\mathcal{E}_D^m = \frac{1}{2} \int_0^L ES u_{,x}^2 dx$$

$$\mathcal{E}_D^f = \frac{1}{2} \int_0^L EI \chi^2 dx = \frac{1}{2} \int_0^L EI \theta_{,x}^2 dx = \frac{1}{2} \int_0^L EI v_{,xxx}^2 dx$$

$$\mathcal{E}_D^{ct} = 0$$

Hypothèse de Timoshenko

L'hypothèse de Timoshenko prend en compte la déformation transverse ϵ_{xy} . Ce modèle est valable pour les poutres dites "épaisses".

X.2 Élément fini de traction-compression : élément barre

On ne s'intéresse dans ce paragraphe qu'à la partie traction-compression de l'énergie de déformation, soit

$$\mathcal{E}_D^m = \frac{1}{2} \int_0^L ES e^2 dx.$$

X.2.1 Espace discrétisé

On construit l'espace $\mathcal{V}_{u_e}^h$ correspondant au déplacement axial $u(x)$ dans un élément à partir de 2 fonctions de forme telles que le déplacement axial discrétisé $u^h(x) \in \mathcal{V}_{u_e}^h$ s'écrive sous la forme

$$u^h(x) = \varphi_a(x)u_a + \varphi_b(x)u_b$$

À partir de la base canonique polynomiale $[1 \ x]$, on montre que les fonctions de forme φ_a et φ_b ont pour expressions :

$$\varphi_a = 1 - \frac{x}{L} \quad \text{et} \quad \varphi_b = \frac{x}{L}$$

On a alors dans l'élément :

$$u^h(x) = [\varphi_a \ \varphi_b] \begin{bmatrix} u_a \\ u_b \end{bmatrix} = \Phi_e \underline{U}_e$$

X.2.2 Opérateur gradient discrétisé

La partie membrane des déformations e s'écrit en fonction du déplacement u comme $e = u_{,x}$ soit aussi

$$e = [\varphi_{a,x} \ \varphi_{b,x}] \begin{bmatrix} u_a \\ u_b \end{bmatrix}$$

En posant \mathbf{B}_m l'opérateur gradient discrétisé pour la partie traction-compression tel que

$$\mathbf{B}_m = [\varphi_{a,x} \ \varphi_{b,x}] = \left[-\frac{1}{L} \ \frac{1}{L} \right]$$

on a

$$e = \mathbf{B}_m \underline{U}_e$$

X.2.3 Matrice de rigidité

En remplaçant dans l'énergie de déformation, il vient :

$$\mathcal{E}_D^m = \frac{1}{2} \int_0^L ES e^2 dx = \frac{1}{2} \int_0^L ES \underline{U}_e^T \mathbf{B}_m^T \mathbf{B}_m \underline{U}_e dx.$$

Soit aussi

$$\mathcal{E}_D^m = \frac{1}{2} \underline{U}_e^T \int_0^L ES \mathbf{B}_m^T \mathbf{B}_m dx \underline{U}_e.$$

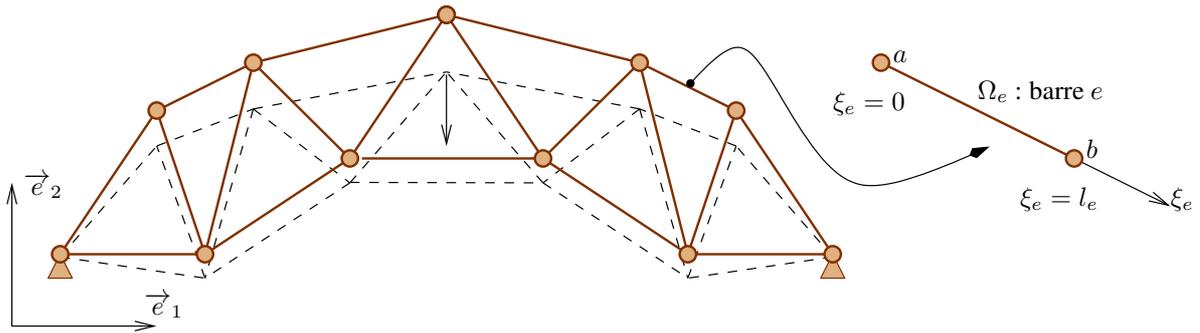


Figure X.4 – Structure de type treillis de barres

En posant la matrice de rigidité de traction-compression

$$\mathbf{K}_m^e = \int_0^L ES \mathbf{B}_m^T \mathbf{B}_m dx$$

on a

$$\mathcal{E}_D^m = \frac{1}{2} \underline{U}_e^T \mathbf{K}_m^e \underline{U}_e.$$

En remplaçant \mathbf{B}_m par son expression, on a :

$$\mathbf{B}_m^T \mathbf{B}_m = \frac{1}{L^2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

En intégrant de 0 à L , on a alors la matrice de rigidité de la partie traction-compression qui vaut

$$\mathbf{K}_m^e = \frac{ES}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

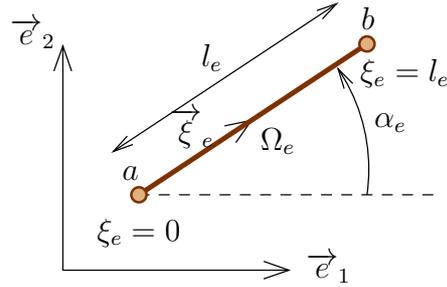
X.2.4 Calcul de treillis de barres

Un treillis est une structure composée de poutres que l'on suppose rotulées entre elles (Fig. X.4). Les barres sont connectées entre elles par des nœuds, centres des liaisons rotules. Les articulations sont supposées parfaites. Cette simplification permet de résoudre relativement facilement le problème. Même si les liaisons ne sont pas réellement des rotules mais des liaisons boulonnées, on peut considérer dans une première approche la structure comme un treillis de barres rotulées. Cela permet de trouver une bonne approximation des efforts normaux dans les barres.

Une barre rotulée à ses deux extrémités ne subit que de la traction-compression. Chaque barre reste un segment de droite après déformation, la figure X.4 montre la déformée globale d'un treillis (barres en tirets).

Dans le cadre de ce cours, on se place dans le cas des treillis plans afin de simplifier les développements.

Un élément e est caractérisé par ses 2 nœuds a et b , par sa longueur l_e et par l'angle α_e qu'il fait avec \vec{e}_1 . On attache à l'élément un repère local $\vec{\xi}_e = \frac{\vec{ab}}{\|\vec{ab}\|}$.



L'énergie de déformation d'une barre ne dépend que des composantes axiales des déplacements des nœuds :

$$\mathcal{E}_D^m = \frac{1}{2} \underline{U}_e^T \mathbf{K}_m^e \underline{U}_e.$$

avec

$$\mathbf{K}_m^e = \frac{ES}{l_e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

et

$$\underline{U}_e = \begin{bmatrix} \vec{u}(\text{nœud } a) \cdot \vec{\xi}_e \\ \vec{u}(\text{nœud } b) \cdot \vec{\xi}_e \end{bmatrix}$$

Or en notant

$$\vec{u}(\text{nœud } a) = U_a \vec{e}_1 + V_a \vec{e}_2$$

on a

$$\vec{u}(\text{nœud } a) \cdot \vec{\xi}_e = U_a \vec{e}_1 \cdot \vec{\xi}_e + V_a \vec{e}_2 \cdot \vec{\xi}_e = U_a \cos \alpha_e + V_a \sin \alpha_e$$

De même, on a :

$$\vec{u}(\text{nœud } b) \cdot \vec{\xi}_e = U_b \cos \alpha_e + V_b \sin \alpha_e$$

Sous forme matricielle, on a :

$$\begin{bmatrix} \vec{u}(\text{nœud } a) \cdot \vec{\xi}_e \\ \vec{u}(\text{nœud } b) \cdot \vec{\xi}_e \end{bmatrix} = \begin{bmatrix} \cos \alpha_e & \sin \alpha_e & 0 & 0 \\ 0 & 0 & \cos \alpha_e & \sin \alpha_e \end{bmatrix} \begin{bmatrix} U_a \\ V_a \\ U_b \\ V_b \end{bmatrix}$$

En posant la matrice de passage \mathbf{T}_e permettant de passer des déplacements des nœuds écrits dans le repère global aux déplacements des nœuds écrits dans le repère local :

$$\mathbf{T}_e = \begin{bmatrix} \cos \alpha_e & \sin \alpha_e & 0 & 0 \\ 0 & 0 & \cos \alpha_e & \sin \alpha_e \end{bmatrix}$$

et en notant $\underline{U}_e = [U_a \ V_a \ U_b \ V_b]^T$ les déplacements des nœuds dans le repère global, on a

$$\begin{bmatrix} \vec{u}(\text{nœud } a) \cdot \vec{\xi}_e \\ \vec{u}(\text{nœud } b) \cdot \vec{\xi}_e \end{bmatrix} = \mathbf{T}_e \underline{U}_e$$

En remplaçant dans l'énergie de déformation, on a :

$$\mathcal{E}_D^m = \frac{1}{2} \underline{U}_e^T \mathbf{T}_e^T \mathbf{K}_m^e \mathbf{T}_e \underline{U}_e.$$

où \mathbf{K}_m^e est la matrice de rigidité de la partie axiale de la poutre développée précédemment :

$$\mathbf{K}_m^e = \frac{ES}{l_e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Le calcul de $\mathbf{T}_e^T \mathbf{K}_m^e \mathbf{T}_e$ donne la matrice de rigidité de l'élément barre dans le plan (x, y) :

$$\mathbf{K}_{\text{barre}}^e = \frac{ES}{l_e} \begin{bmatrix} \cos^2 \alpha_e & \cos \alpha_e \sin \alpha_e & -\cos^2 \alpha_e & -\cos \alpha_e \sin \alpha_e \\ \cos \alpha_e \sin \alpha_e & \sin^2 \alpha_e & -\cos \alpha_e \sin \alpha_e & -\sin^2 \alpha_e \\ -\cos^2 \alpha_e & -\cos \alpha_e \sin \alpha_e & \cos^2 \alpha_e & \cos \alpha_e \sin \alpha_e \\ -\cos \alpha_e \sin \alpha_e & -\sin^2 \alpha_e & \cos \alpha_e \sin \alpha_e & \sin^2 \alpha_e \end{bmatrix}$$

L'énergie de déformation de l'élément barre vaut alors

$$\mathcal{E}_D^m = \frac{1}{2} \underline{U}_e^T \mathbf{K}_{\text{barre}}^e \underline{U}_e.$$

X.3 Éléments finis de flexion de type Euler-Bernoulli

On ne s'intéresse dans ce paragraphe qu'à la partie flexion de l'énergie de déformation avec l'hypothèse d'Euler-Bernoulli, soit

$$\mathcal{E}_D^f = \frac{1}{2} \int_0^L EI \chi^2 dx = \frac{1}{2} \int_0^L EI v_{,xx}^2 dx.$$

X.3.1 Espace discrétisé

On construit l'espace $\mathcal{V}_{\mathbf{v}_e}^h$ dans un élément à partir de 4 fonctions de forme $\varphi_1, \varphi_2, \varphi_3$ et φ_4 telles que toute fonction v^h et $\mathcal{V}_{\mathbf{v}_e}^h$ s'écrive sous la forme :

$$v^h(x) = \varphi_1(x)v_a + \varphi_2(x)\theta_a + \varphi_3(x)v_b + \varphi_4(x)\theta_b$$

où v_a et v_b sont les déplacements suivant \vec{y} des deux nœuds et θ_a et θ_b sont les rotations des sections aux deux nœuds. On a alors dans un élément :

$$v^h(x) = \mathbf{\Phi}_e \underline{V}_e$$

avec

$$\mathbf{\Phi}_e = [\varphi_1(x) \ \varphi_2(x) \ \varphi_3(x) \ \varphi_4(x)]$$

et

$$\underline{V}_e = [v_a \ \theta_a \ v_b \ \theta_b]^T.$$

On définit \mathcal{V}_e^h comme

$$\mathcal{V}_e^h = \text{vect} \{ 1, x, x^2, x^3 \} = \text{vect} \{ \varphi_1(x), \varphi_2(x), \varphi_3(x), \varphi_4(x) \}$$

En prenant successivement les 4 termes de la base canonique polynomiale de \mathcal{V}_e^h , on a :

$$\text{— pour } u^h(x) = 1; u^{h'}(x) = 0$$

$$u^h(0) = 1, u^h(L) = 1$$

$$\varphi_1(x) + \varphi_3(x) = 1$$

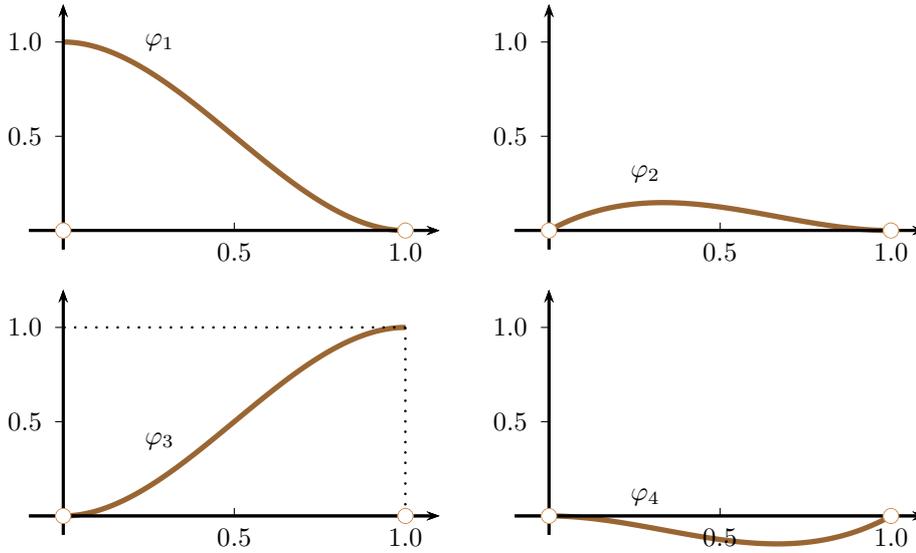


Figure X.5 – Fonctions de forme d’Hermite de l’élément poutre d’Euler-Bernoulli pour $L = 1$.

- pour $u^h(x) = x : u^{h'}(x) = 1$
 $u^h(0) = 0, u^h(L) = L$

$$\varphi_2(x) + \varphi_3(x)L + \varphi_4(x) = x$$

- pour $u^h(x) = x^2 : u^{h'}(x) = 2x$
 $u^h(0) = 0, u^{h'}(0) = 0, u^h(L) = L^2, u^{h'}(L) = 2L$

$$\varphi_3(x)L^2 + \varphi_4(x)2L = x^2$$

- pour $u^h(x) = x^3 : u^{h'}(x) = 3x^2$
 $u^h(0) = 0, u^{h'}(0) = 0, u^h(L) = L^3, u^{h'}(L) = 3L^2$

$$\varphi_3(x)L^3 + \varphi_4(x)3L^2 = x^3$$

Ces 4 équations s’écrivent sous forme d’un système à 4 inconnues :

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & L & 1 \\ 0 & 0 & L^2 & 2L \\ 0 & 0 & L^3 & 3L^2 \end{bmatrix} \begin{bmatrix} \varphi_1(x) \\ \varphi_2(x) \\ \varphi_3(x) \\ \varphi_4(x) \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix}$$

La solution de ce système est

$$\varphi_1(x) = 1 - \frac{3}{L^2}x^2 + \frac{2}{L^3}x^3 ; \quad \varphi_2(x) = x - \frac{2}{L}x^2 + \frac{1}{L^2}x^3$$

$$\varphi_3(x) = \frac{3}{L^2}x^2 - \frac{2}{L^3}x^3 ; \quad \varphi_4(x) = -\frac{1}{L}x^2 + \frac{1}{L^2}x^3$$

Les fonctions φ_i sont des polynômes de degrés 3, ce sont des polynômes de type Hermite (Fig. X.5).

X.3.2 Opérateur gradient discrétisé

La courbure χ s'exprime en fonction du déplacement transversal v comme

$$\chi = -\frac{\partial^2 v}{\partial x^2},$$

soit pour un champ de déplacement discrétisé v^h :

$$\chi = -\frac{\partial^2}{\partial x^2} \mathbf{\Phi}_e \underline{V}_e.$$

En posant \mathbf{B}_f l'opérateur gradient discrétisé pour la partie flexion tel que

$$\mathbf{B}_f = \frac{\partial^2}{\partial x^2} \mathbf{\Phi}_e = [\varphi_1''(x) \quad \varphi_2''(x) \quad \varphi_3''(x) \quad \varphi_4''(x)],$$

on a

$$\chi = -\mathbf{B}_f \underline{V}_e.$$

Les dérivées secondes des fonctions de forme valent :

$$\begin{aligned} \varphi_1''(x) &= \frac{2}{L} \left(-\frac{3}{L} + \frac{6}{L^2}x \right); & \varphi_2''(x) &= \frac{2}{L} \left(-2 + \frac{3}{L}x \right) \\ \varphi_3''(x) &= \frac{2}{L} \left(\frac{3}{L} - \frac{6}{L^2}x \right); & \varphi_4''(x) &= \frac{2}{L} \left(-1 + \frac{3}{L}x \right). \end{aligned}$$

X.3.3 Matrice de rigidité

Le calcul de la matrice de rigidité pour la partie flexion se fait en calculant l'énergie de déformation de flexion associée à un champ de déplacement discrétisé v^h :

$$\mathcal{E}_D^f = \frac{1}{2} \underline{V}_e^T EI \underbrace{\int_0^L \mathbf{B}_f^T \mathbf{B}_f dx}_{=\mathbf{K}_f^e} \underline{V}_e.$$

L'expression de \mathbf{K}_f^e est :

$$\mathbf{K}_f^e = EI \int_0^L \begin{bmatrix} \varphi_1''(x)\varphi_1''(x) & \varphi_1''(x)\varphi_2''(x) & \varphi_1''(x)\varphi_3''(x) & \varphi_1''(x)\varphi_4''(x) \\ \varphi_2''(x)\varphi_1''(x) & \varphi_2''(x)\varphi_2''(x) & \varphi_2''(x)\varphi_3''(x) & \varphi_2''(x)\varphi_4''(x) \\ \varphi_3''(x)\varphi_1''(x) & \varphi_3''(x)\varphi_2''(x) & \varphi_3''(x)\varphi_3''(x) & \varphi_3''(x)\varphi_4''(x) \\ \varphi_4''(x)\varphi_1''(x) & \varphi_4''(x)\varphi_2''(x) & \varphi_4''(x)\varphi_3''(x) & \varphi_4''(x)\varphi_4''(x) \end{bmatrix} dx$$

La matrice de rigidité de l'élément vaut pour la partie flexion :

$$\mathbf{K}_f^e = \frac{2EI}{L^3} \begin{bmatrix} 6 & 3L & -6 & 3L \\ 3L & 2L^2 & -3L & L^2 \\ -6 & -3L & 6 & -3L \\ 3L & L^2 & -3L & 2L^2 \end{bmatrix}$$

XI — Élément fini de plaque

XI.1 Modèle de plaque

XI.1.1 Modèle géométrique

Une plaque est un domaine dont une dimension, l'épaisseur notée h , est plus petite que les deux autres (Fig. XI.1). Le plan moyen (noté S) de la plaque a pour repère (\vec{x}, \vec{y}) . La troisième direction \vec{z} est celle de l'épaisseur. Le domaine Ω de la plaque peut être défini comme :

$$\Omega = S \times \left[-\frac{h}{2}, \frac{h}{2} \right]$$

où un point $M \in \Omega$ a pour coordonnées (x, y, z) avec $(x, y) \in S$ et $z \in [-\frac{h}{2}, \frac{h}{2}]$. La projection de M sur le plan moyen S est notée $m(x, y)$.

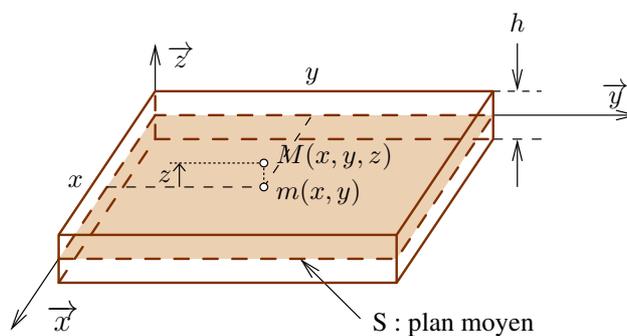


Figure XI.1 – Structure de type plaque.

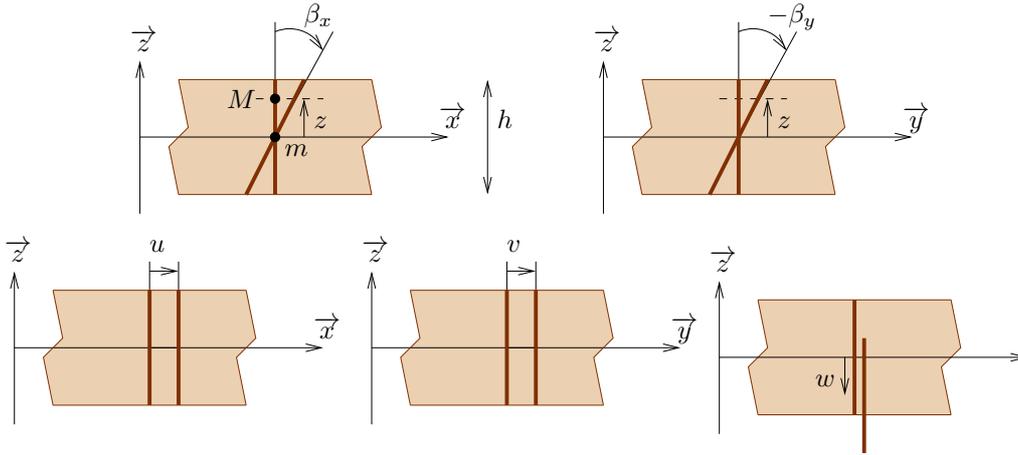


Figure XI.2 – Cinématique de plaque.

XI.1.2 Hypothèse cinématique

On appelle "segment" un segment de droite initialement perpendiculaire à la surface moyenne de la plaque en un point m . On fait l'hypothèse que ce segment reste droit après déformation, mais pas nécessairement perpendiculaire à la surface moyenne. Autrement dit, chaque segment a une cinématique de corps rigide qui peut être décomposée en 2 effets (Fig. XI.2) :

- déplacement suivant \vec{x} et \vec{y} (effet membrane),
- rotation des segments autour du plan moyen et déplacement suivant \vec{z} (effet flexion).

Le déplacement de m est noté $\vec{u}(m)$:

$$\vec{u}(m) = u(x, y)\vec{x} + v(x, y)\vec{y} + w(x, y)\vec{z}.$$

La rotation d'un segment autour de \vec{x} est notée $-\beta_y$, la rotation autour de \vec{y} est notée β_x , soit la rotation $\vec{\Omega} = -\beta_y\vec{x} + \beta_x\vec{y}$.

En supposant que les déplacements et les rotations sont petits, le déplacement de M s'écrit

$$\vec{u}(M) = \vec{u}(m) + \overrightarrow{Mm} \wedge \vec{\Omega} = u(x, y)\vec{x} + v(x, y)\vec{y} + w(x, y)\vec{z} - z\vec{z} \wedge (-\beta_y(x, y)\vec{x} + \beta_x(x, y)\vec{y})$$

Le champ de déplacement de la plaque est alors $\vec{u}(M)$ tel que

$$\forall (x, y) \in S ; \quad \forall z \in \left[-\frac{h}{2}, \frac{h}{2}\right],$$

$$\vec{u}(x, y, z) = (u(x, y) + z\beta_x(x, y))\vec{x} + (v(x, y) + z\beta_y(x, y))\vec{y} + w(x, y)\vec{z}$$

ou encore

$$\vec{u}(M) = \begin{bmatrix} u(x, y) + z\beta_x(x, y) \\ v(x, y) + z\beta_y(x, y) \\ w(x, y) \end{bmatrix}_{(\vec{x}, \vec{y}, \vec{z})}$$

avec $u, v, w, \beta_x, \beta_y \in \mathcal{V}$. Ce dernier ensemble \mathcal{V} est l'ensemble des champs de fonctions "suffisamment régulières" définis de S vers \mathbb{R} :

$$\mathcal{V} = \left\{ \begin{array}{l} f \text{ "réguliers"} : S \rightarrow \mathbb{R} \\ x, y \mapsto f(x, y) \end{array} \right\}$$

XI.1.3 Écriture de la relation déformations-déplacements

On remarque que l'hypothèse cinématique précédente conduit à $\epsilon_{zz} = 0$, on note alors les déformations sous la forme

$$\underline{\epsilon} = \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ 2\epsilon_{xy} \\ 2\epsilon_{xz} \\ 2\epsilon_{yz} \end{bmatrix}.$$

On sépare les déformations en 2 parties :

— partie plane des déformations :

$$\begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ 2\epsilon_{xy} \end{bmatrix} = \underbrace{\begin{bmatrix} u_{,x} \\ v_{,y} \\ u_{,y} + v_{,x} \end{bmatrix}}_{=\underline{\epsilon}} + z \underbrace{\begin{bmatrix} \beta_{x,x} \\ \beta_{y,y} \\ \beta_{x,y} + \beta_{y,x} \end{bmatrix}}_{=\underline{\chi}}$$

où $\underline{\epsilon}$ caractérise les effets membrane et $\underline{\chi}$ caractérise les effets flexion (courbures),

— partie hors plan des déformations : déformations transverses

$$\begin{bmatrix} 2\epsilon_{xz} \\ 2\epsilon_{yz} \end{bmatrix} = \underbrace{\begin{bmatrix} \beta_x + w_{,x} \\ \beta_y + w_{,y} \end{bmatrix}}_{=\underline{\gamma}}.$$

XI.1.4 Relation de comportement entre les déformations et les contraintes

Une plaque étant un milieu mince, on suppose que la contrainte σ_{zz} est nulle dans toute la plaque. Ceci est en contradiction avec l'hypothèse cinématique qui donne $\epsilon_{zz} = 0$, mais le modèle ainsi défini donne malgré tout de bons résultats. À partir de la loi de Hooke

$$\underline{\epsilon} = \frac{1+\nu}{E} \underline{\sigma} - \frac{\nu}{E} \text{Tr}(\underline{\sigma}) \mathbf{1},$$

les déformations s'écrivent en fonction des contraintes comme :

$$\begin{aligned} \epsilon_{xx} &= \frac{1+\nu}{E} \sigma_{xx} - \frac{\nu}{E} (\sigma_{xx} + \sigma_{yy}) \\ \epsilon_{yy} &= \frac{1+\nu}{E} \sigma_{yy} - \frac{\nu}{E} (\sigma_{xx} + \sigma_{yy}) \\ \epsilon_{zz} &= -\frac{\nu}{E} (\sigma_{xx} + \sigma_{yy}) ; \quad \epsilon_{xy} = \frac{1+\nu}{E} \sigma_{xy} ; \quad \epsilon_{xz} = \frac{1+\nu}{E} \sigma_{xz} ; \quad \epsilon_{yz} = \frac{1+\nu}{E} \sigma_{yz}. \end{aligned}$$

En inversant ces relations, on a

$$\underline{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \\ \sigma_{xz} \\ \sigma_{yz} \end{bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 & 0 & 0 \\ \nu & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{1-\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1-\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ 2\epsilon_{xy} \\ 2\epsilon_{xz} \\ 2\epsilon_{yz} \end{bmatrix} = \underline{C} \underline{\epsilon}.$$

On sépare les contraintes en deux parties comme pour les déformations :

— contraintes dans le plan :

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix}$$

— contraintes hors plan : cisaillement transverse

$$\begin{bmatrix} \sigma_{xz} \\ \sigma_{yz} \end{bmatrix}$$

Avec ces notations, les relations entre les contraintes et les déformations s'écrivent :

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} = \frac{E}{1-\nu^2} \underbrace{\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}}_{\mathbf{C}_1} \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ 2\epsilon_{xy} \end{bmatrix}$$

et

$$\begin{bmatrix} \sigma_{xz} \\ \sigma_{yz} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{E}{2(1+\nu)} & 0 \\ 0 & \frac{E}{2(1+\nu)} \end{bmatrix}}_{\mathbf{C}_2} \begin{bmatrix} 2\epsilon_{xz} \\ 2\epsilon_{yz} \end{bmatrix}$$

XI.1.5 Énergie de déformation

Avec les hypothèses précédentes, l'énergie de déformation vaut :

$$\mathcal{E}_D = \frac{1}{2} \int_{\Omega} \underline{\sigma}^T \underline{\epsilon} \, d\Omega = \frac{1}{2} \int_{\Omega} [(\underline{e} + z\underline{\chi})^T \quad \underline{\gamma}^T] \begin{bmatrix} \mathbf{C}_1 & 0 \\ 0 & \mathbf{C}_2 \end{bmatrix} \begin{bmatrix} \underline{e} + z\underline{\chi} \\ \underline{\gamma} \end{bmatrix} \, d\Omega$$

L'intégrale dans Ω est décomposée en une intégrale sur la surface moyenne S et une intégrale dans l'épaisseur suivant z :

$$\mathcal{E}_D = \frac{1}{2} \int_S \int_{-\frac{h}{2}}^{\frac{h}{2}} \left\{ \underline{e}^T \mathbf{C}_1 \underline{e} + z^2 \underline{\chi}^T \mathbf{C}_1 \underline{\chi} + z \underline{\chi}^T \mathbf{C}_1 \underline{e} + z \underline{e}^T \mathbf{C}_1 \underline{\chi} + \underline{\gamma}^T \mathbf{C}_2 \underline{\gamma} \right\} \, dz \, dS$$

Dans le cas homogène isotrope, les matrices de comportement \mathbf{C}_1 et \mathbf{C}_2 sont constantes dans l'épaisseur et ne dépendent donc pas de z , ce qui n'est pas le cas pour les matériaux composites. Les intégrales dans l'épaisseur valent :

$$\int_{-\frac{h}{2}}^{\frac{h}{2}} dz = h ; \quad \int_{-\frac{h}{2}}^{\frac{h}{2}} z \, dz = 0 ; \quad \int_{-\frac{h}{2}}^{\frac{h}{2}} z^2 \, dz = \frac{h^3}{12}.$$

L'énergie de déformation vaut alors

$$\mathcal{E}_D = \frac{1}{2} \int_S \left\{ h \underline{e}^T \tilde{\mathbf{C}} \underline{e} + \frac{h^3}{12} \underline{\chi}^T \mathbf{C}_1 \underline{\chi} + h \underline{\gamma}^T \mathbf{C}_2 \underline{\gamma} \right\} \, dS$$

On remarque que les termes de couplage entre $\underline{\chi}$ et \underline{e} disparaissent dans le cas d'un matériau homogène et isotrope, contrairement à un matériau composite pour lequel il y a un couplage. Finalement, en posant les matrices de comportement suivantes intégrées dans l'épaisseur :

$$\mathbf{C}_m = h \mathbf{C}_1 = \frac{Eh}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix},$$

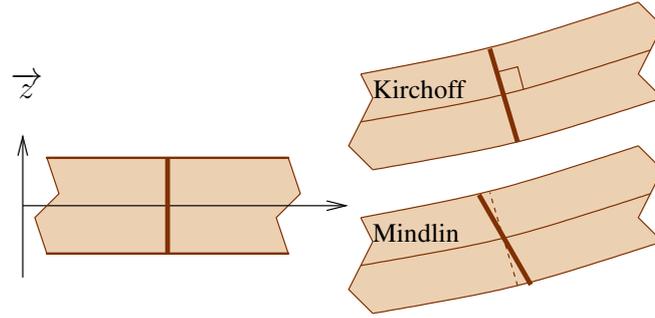


Figure XI.3 – Différences entre les modèles cinématiques de Kirchoff-Love et Reissner-Mindlin

$$\mathbf{C}_f = \frac{h^3}{12} \mathbf{C}_1 = \frac{Eh^3}{12(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix},$$

$$\mathbf{C}_c = h\mathbf{C}_2 = \frac{Eh}{2(1+\nu)} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

l'énergie de déformation s'écrit sous la forme :

$$\mathcal{E}_D = \frac{1}{2} \int_S \underline{\underline{\epsilon}}^T \mathbf{C}_m \underline{\underline{\epsilon}} \, dS + \frac{1}{2} \int_S \underline{\underline{\chi}}^T \mathbf{C}_f \underline{\underline{\chi}} \, dS + \frac{1}{2} \int_S \underline{\underline{\gamma}}^T \mathbf{C}_c \underline{\underline{\gamma}} \, dS.$$

L'énergie de déformation peut être séparée en 3 contributions :

$$\begin{aligned} \mathcal{E}_D^m &= \frac{1}{2} \int_S \underline{\underline{\epsilon}}^T \mathbf{C}_m \underline{\underline{\epsilon}} \, dS, \text{ énergie de déformation de membrane,} \\ \mathcal{E}_D^f &= \frac{1}{2} \int_S \underline{\underline{\chi}}^T \mathbf{C}_f \underline{\underline{\chi}} \, dS, \text{ énergie de déformation de flexion,} \\ \mathcal{E}_D^{ct} &= \frac{1}{2} \int_S \underline{\underline{\gamma}}^T \mathbf{C}_c \underline{\underline{\gamma}} \, dS, \text{ énergie de déformation de cisaillement transverse.} \end{aligned}$$

On a alors :

$$\mathcal{E}_D = \mathcal{E}_D^m + \mathcal{E}_D^f + \mathcal{E}_D^{ct}.$$

XI.1.6 Hypothèses sur les déformations transverses

Deux hypothèses sont classiquement utilisées pour les plaques ; Kirchoff-Love ou Reissner-Mindlin (Fig. XI.3).

L'hypothèse de Kirchoff-Love est l'équivalent de celle d'Euler-Bernoulli pour les poutres, on suppose que les déformations transverses sont nulles :

$$\epsilon_{xz} = \epsilon_{yz} = 0.$$

Ceci entraîne que

$$\beta_x = -w_{,x} ; \beta_y = -w_{,y} \text{ et } \mathcal{E}_D^{ct} = 0.$$

D'un point de vue cinématique cela veut dire qu'un segment initialement droit et perpendiculaire à la surface moyenne le reste après déformation. D'un point de vue éléments finis, il faut alors qu'il y ait compatibilité entre w et les rotations β_x et β_y . L'espace d'approximation de w est généralement construit avec des polynômes de degré 3 et les fonctions de forme sont des polynômes d'Hermite. Ce modèle est adapté aux plaques élancées et minces.

L'hypothèse de Reissner-Mindlin est l'équivalent de celle de Timoshenko pour les poutres, on prend en compte le cisaillement transverse. Ce modèle est adapté aux plaques peu élancées et plutôt épaisses.

XI.2 Élément fini de type Reissner-Mindlin

XI.2.1 Espace discrétisé

On définit l'espace \mathcal{V}^h comme l'espace des fonctions à 2 variables engendré par les fonctions de forme φ_i où i varie de 1 au nombre de nœuds n du maillage :

$$\mathcal{V}^h = Vect \{ \varphi_1, \varphi_2, \dots, \varphi_n \}$$

Le champ de déplacements éléments finis \vec{v}^h est alors écrit comme :

$$\forall (x, y) \in S ; \quad \forall z \in \left[-\frac{h}{2}, \frac{h}{2} \right],$$

$$\vec{v}^h(x, y, z) = (u^h(x, y) + z\beta_x^h(x, y)) \vec{x} + (v^h(x, y) + z\beta_y^h(x, y)) \vec{y} + w^h(x, y) \vec{z}$$

avec $u^h, v^h, w^h, \beta_x^h, \beta_y^h \in \mathcal{V}^h$. Autrement dit, les déplacements et les rotations sont approximés par les mêmes fonctions de forme. En introduisant la matrice des fonctions de forme Φ suivante :

$$\Phi = [\varphi_1, \varphi_2, \dots, \varphi_n]$$

on a alors :

$$u^h(x, y) = \sum_i^n \varphi_i(x, y) u_i = \Phi \underline{U} ; \quad v^h(x, y) = \sum_i^n \varphi_i(x, y) v_i = \Phi \underline{V}$$

$$w^h(x, y) = \sum_i^n \varphi_i(x, y) w_i = \Phi \underline{W} ; \quad \beta_x^h(x, y) = \sum_i^n \varphi_i(x, y) \beta_i^x = \Phi \underline{\beta}_x$$

$$\beta_y^h(x, y) = \sum_i^n \varphi_i(x, y) \beta_i^y = \Phi \underline{\beta}_y$$

où $\underline{U}, \underline{V}, \underline{W}, \underline{\beta}_x$ et $\underline{\beta}_y$ sont respectivement les déplacements et rotations aux nœuds du maillage. Le calcul au niveau élémentaire utilise toute la stratégie du calcul par éléments finis détaillée dans les chapitres précédents. Aucune forme d'élément n'est imposée, ceci est valable pour les triangles et les quadrangles.

XI.2.2 Opérateur gradient discrétisé

La partie plane $\underline{\epsilon}$ des déformations vaut pour un champ de déplacement \vec{v}^h de \mathcal{V}^h :

$$\underline{\epsilon}(\vec{v}^h) = \begin{bmatrix} u^h_{,x} \\ v^h_{,y} \\ u^h_{,y} + v^h_{,x} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} u^h \\ v^h \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} \Phi & 0 \\ 0 & \Phi \end{bmatrix} \begin{bmatrix} \underline{U} \\ \underline{V} \end{bmatrix}$$

Soit

$$\underline{\epsilon}(\vec{v}^h) = \begin{bmatrix} \frac{\partial}{\partial x} \Phi & 0 \\ 0 & \frac{\partial}{\partial y} \Phi \\ \frac{\partial}{\partial y} \Phi & \frac{\partial}{\partial x} \Phi \end{bmatrix} \begin{bmatrix} \underline{U} \\ \underline{V} \end{bmatrix}.$$

En posant \mathbf{B}_m l'opérateur gradient discrétisé pour la partie membrane tel que

$$\mathbf{B}_m = \begin{bmatrix} \frac{\partial}{\partial x} \Phi & 0 \\ 0 & \frac{\partial}{\partial y} \Phi \\ \frac{\partial}{\partial y} \Phi & \frac{\partial}{\partial x} \Phi \end{bmatrix}$$

on a

$$\underline{\epsilon}(\vec{v}^h) = \mathbf{B}_m \begin{bmatrix} \underline{U} \\ \underline{V} \end{bmatrix}.$$

La partie courbure $\underline{\chi}$ des déformations vaut pour un champ de déplacement \vec{v}^h de \mathcal{V}^h :

$$\underline{\chi}(\vec{v}^h) = \begin{bmatrix} \beta_{x,x}^h \\ \beta_{y,y}^h \\ \beta_{x,y}^h + \beta_{y,x}^h \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} \beta_x^h \\ \beta_y^h \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} \Phi & 0 \\ 0 & \Phi \end{bmatrix} \begin{bmatrix} \underline{\beta}_x \\ \underline{\beta}_y \end{bmatrix}.$$

Soit

$$\underline{\chi} = \begin{bmatrix} \frac{\partial}{\partial x} \Phi & 0 \\ 0 & \frac{\partial}{\partial y} \Phi \\ \frac{\partial}{\partial y} \Phi & \frac{\partial}{\partial x} \Phi \end{bmatrix} \begin{bmatrix} \underline{\beta}_x \\ \underline{\beta}_y \end{bmatrix}.$$

En posant \mathbf{B}_f l'opérateur gradient discrétisé pour la partie courbure tel que

$$\mathbf{B}_f = \begin{bmatrix} \frac{\partial}{\partial x} \Phi & 0 \\ 0 & \frac{\partial}{\partial y} \Phi \\ \frac{\partial}{\partial y} \Phi & \frac{\partial}{\partial x} \Phi \end{bmatrix}$$

on a

$$\underline{\chi}(\vec{v}^h) = \mathbf{B}_f \begin{bmatrix} \underline{\beta}_x \\ \underline{\beta}_y \end{bmatrix}.$$

La partie déformations transverses $\underline{\gamma}$ des déformations vaut pour un champ de déplacement \vec{v}^h de \mathcal{V}^h :

$$\underline{\gamma}(\vec{v}^h) = \begin{bmatrix} \beta_x^h + w_{,x}^h \\ \beta_y^h + w_{,y}^h \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 1 & 0 \\ \frac{\partial}{\partial y} & 0 & 1 \end{bmatrix} \begin{bmatrix} w^h \\ \beta_x^h \\ \beta_y^h \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 1 & 0 \\ \frac{\partial}{\partial y} & 0 & 1 \end{bmatrix} \begin{bmatrix} \Phi & 0 & 0 \\ 0 & \Phi & 0 \\ 0 & 0 & \Phi \end{bmatrix} \begin{bmatrix} \underline{W} \\ \underline{\beta}_x \\ \underline{\beta}_y \end{bmatrix}.$$

Soit

$$\underline{\gamma} = \begin{bmatrix} \frac{\partial}{\partial x} \Phi & \Phi & 0 \\ \frac{\partial}{\partial y} \Phi & 0 & \Phi \end{bmatrix} \begin{bmatrix} \underline{W} \\ \underline{\beta}_x \\ \underline{\beta}_y \end{bmatrix}.$$

En posant \mathbf{B}_{ct} l'opérateur gradient discrétisé pour la partie déformations transverses tel que

$$\mathbf{B}_{ct} = \begin{bmatrix} \frac{\partial}{\partial x} \Phi & \Phi & 0 \\ \frac{\partial}{\partial y} \Phi & 0 & \Phi \end{bmatrix}$$

on a

$$\underline{\gamma}(\vec{v}^h) = \mathbf{B}_{ct} \begin{bmatrix} \underline{W} \\ \underline{\beta}_x \\ \underline{\beta}_y \end{bmatrix}.$$

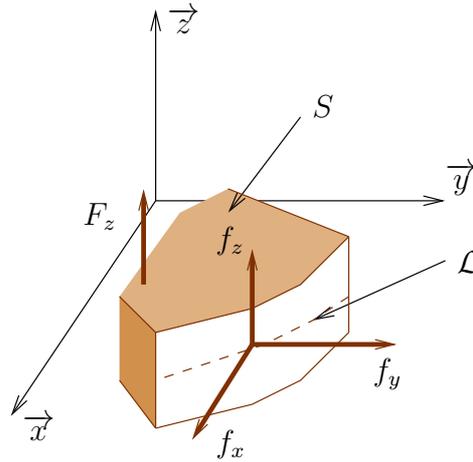


Figure XI.4 – Forces extérieures appliquées sur la plaque.

XI.2.3 Matrice de rigidité

En remplaçant dans l'énergie de déformation, il vient

$$\begin{aligned} \mathcal{E}_D = \frac{1}{2} \int_S \left\{ [\underline{U}^T \quad \underline{V}^T] \mathbf{B}_m^T \mathbf{C}_m \mathbf{B}_m \begin{bmatrix} \underline{U} \\ \underline{V} \end{bmatrix} + [\underline{\beta}_x^T \quad \underline{\beta}_y^T] \mathbf{B}_f^T \mathbf{C}_f \mathbf{B}_f \begin{bmatrix} \underline{\beta}_x \\ \underline{\beta}_y \end{bmatrix} \right. \\ \left. + [\underline{W}^T \quad \underline{\beta}_x^T \quad \underline{\beta}_y^T] \mathbf{B}_{ct}^T \mathbf{C}_{ct} \mathbf{B}_{ct} \begin{bmatrix} \underline{W} \\ \underline{\beta}_x \\ \underline{\beta}_y \end{bmatrix} \right\} dS \end{aligned}$$

$$\mathcal{E}_D = \frac{1}{2} \underline{Q}^T \int_S \begin{bmatrix} \begin{bmatrix} \mathbf{B}_m^T \mathbf{C}_m \mathbf{B}_m \\ \mathbf{0} \end{bmatrix} & \mathbf{0} \\ \begin{bmatrix} \mathbf{B}_{ct}^T \mathbf{C}_{ct} \mathbf{B}_{ct} \\ \vdots \end{bmatrix} & \begin{bmatrix} \dots \\ +\mathbf{B}_f^T \mathbf{C}_f \mathbf{B}_f \end{bmatrix} \end{bmatrix} dS \underline{Q}$$

avec $\underline{Q} = [\underline{U}^T \quad \underline{V}^T \quad \underline{W}^T \quad \underline{\beta}_x^T \quad \underline{\beta}_y^T]^T$.

La matrice de rigidité de l'élément plaque de Reissner-Mindlin vaut donc :

$$\mathbf{K}^e = \int_S \begin{bmatrix} \begin{bmatrix} \mathbf{B}_m^T \mathbf{C}_m \mathbf{B}_m \\ \mathbf{0} \end{bmatrix} & \mathbf{0} \\ \begin{bmatrix} \mathbf{B}_{ct}^T \mathbf{C}_{ct} \mathbf{B}_{ct} \\ \vdots \end{bmatrix} & \begin{bmatrix} \dots \\ +\mathbf{B}_f^T \mathbf{C}_f \mathbf{B}_f \end{bmatrix} \end{bmatrix} dS.$$

XI.2.4 Forces extérieures généralisées

Les forces extérieures sont appliquées sur le contour de la plaque ainsi que sur la surface moyenne (fig. XI.4). Sur le contour, les forces surfaciques sont notées $[f_x \quad f_y \quad f_z]$. On suppose que la force surfacique appliquée sur la surface moyenne n'est portée que par \vec{z} et vaut F_z . On note s l'abscisse curviligne sur la ligne moyenne \mathcal{L} du contour de la plaque. Le travail des efforts

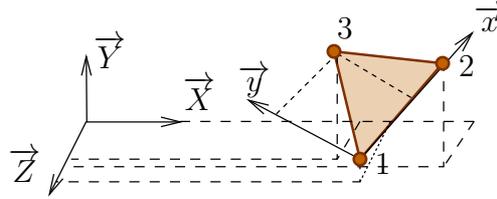


Figure XI.5 – Élément de plaque de type triangle à 3 nœuds dans l'espace.

extérieurs dans un champ de déplacement \vec{v} de la plaque vault

$$\begin{aligned} \mathcal{W}(\vec{v}) &= \int_{\text{contour, surface}} [f_x \ f_y \ f_z] \begin{bmatrix} u(x, y) + z\beta_x(x, y) \\ v(x, y) + z\beta_y(x, y) \\ w(x, y) \end{bmatrix} + \int_S F_z w(x, y) \\ &= \int_{\text{contour, ligne}=\mathcal{L}} \left\{ \underbrace{\int_{-\frac{h}{2}}^{\frac{h}{2}} f_x dz}_{=t_x(s)} u(x, y) + \underbrace{\int_{-\frac{h}{2}}^{\frac{h}{2}} f_y dz}_{=t_y(s)} v(x, y) + \underbrace{\int_{-\frac{h}{2}}^{\frac{h}{2}} f_z dz}_{=t_z(s)} w(x, y) \right. \\ &\quad \left. + \underbrace{\int_{-\frac{h}{2}}^{\frac{h}{2}} z f_x dz}_{=m_x(s)} \beta_x + \underbrace{\int_{-\frac{h}{2}}^{\frac{h}{2}} z f_y dz}_{=m_y(s)} \beta_y \right\} ds + \int_S F_z w(x, y) \end{aligned}$$

où $t_x(s)$, $t_y(s)$ et $t_z(s)$ sont respectivement les efforts linéiques sur le contour de la plaque suivant \vec{x} , \vec{y} et \vec{z} , et où $m_x(s)$ et $m_y(s)$ sont les moments linéiques sur le contour suivant \vec{y} et \vec{x} . Le travail des forces extérieures dans un champ de déplacement \vec{v}^h éléments finis de la plaque vault

$$\begin{aligned} \mathcal{W}(\vec{v}^h) &= \underline{U}^T \int_{\mathcal{L}} t_x(s) \Phi^T ds + \underline{V}^T \int_{\mathcal{L}} t_y(s) \Phi^T ds + \underline{W}^T \int_{\mathcal{L}} t_z(s) \Phi^T ds \\ &\quad + \underline{\beta}_x^T \int_{\mathcal{L}} m_x(s) \Phi^T ds + \underline{\beta}_y^T \int_{\mathcal{L}} m_y(s) \Phi^T ds + \underline{W}^T \int_S F_z \Phi^T dS \end{aligned}$$

En posant le vecteur des forces généralisées suivant :

$$\underline{F} = \begin{bmatrix} \int_{\mathcal{L}} t_x(s) \Phi^T ds \\ \int_{\mathcal{L}} t_y(s) \Phi^T ds \\ \int_{\mathcal{L}} t_z(s) \Phi^T ds + \int_S F_z \Phi^T dS \\ \int_{\mathcal{L}} m_x(s) \Phi^T ds \\ \int_{\mathcal{L}} m_y(s) \Phi^T ds \end{bmatrix},$$

on a

$$\mathcal{W}(\vec{v}^h) = \underline{Q}^T \underline{F}.$$

XI.3 Assemblage de plaques dans l'espace

Les éléments de plaques peuvent être utilisés pour calculer des structures composées de plaques dans l'espace. Ils peuvent aussi être utilisés pour modéliser des coques.

L'élément triangle est utilisé pour illustrer la démarche (Fig. XI.5). Cet élément est placé dans l'espace associé au repère global $(\vec{X}, \vec{Y}, \vec{Z})$. Pour chaque élément on définit un repère local plan $(\vec{x}, \vec{y}, \vec{z})$:

— l'axe \vec{x} est porté par le vecteur $\vec{12}$ reliant le nœud 1 au nœud 2 :

$$\vec{x} = \frac{\vec{12}}{\|\vec{12}\|},$$

— le vecteur \vec{z} est construit de façon à être perpendiculaire au triangle :

$$\vec{z} = \frac{\vec{x} \wedge \vec{13}}{\|\vec{x} \wedge \vec{13}\|},$$

— on en déduit l'axe $\vec{y} = \vec{z} \wedge \vec{x}$.

Le déplacement et la rotation d'un segment dans le repère global $(\vec{X}, \vec{Y}, \vec{Z})$ sont notés :

$$\vec{u} = U_X \vec{X} + U_Y \vec{Y} + U_Z \vec{Z} \quad \text{et} \quad \vec{\Omega} = \theta_X \vec{X} + \theta_Y \vec{Y} + \theta_Z \vec{Z}$$

Le déplacement et la rotation d'un segment dans le repère local $(\vec{x}, \vec{y}, \vec{z})$ sont notés :

$$\vec{u} = u \vec{x} + v \vec{y} + w \vec{z} \quad \text{et} \quad \vec{\Omega} = -\beta_y \vec{x} + \beta_x \vec{y}$$

Avec ces notations, on obtient les relations suivantes :

$$u = \vec{x} \cdot (U_X \vec{X} + U_Y \vec{Y} + U_Z \vec{Z}) \quad ; \quad v = \vec{y} \cdot (U_X \vec{X} + U_Y \vec{Y} + U_Z \vec{Z})$$

$$w = \vec{z} \cdot (U_X \vec{X} + U_Y \vec{Y} + U_Z \vec{Z}) \quad ; \quad \beta_x = \vec{y} \cdot (\theta_X \vec{X} + \theta_Y \vec{Y} + \theta_Z \vec{Z})$$

$$\beta_y = -\vec{x} \cdot (\theta_X \vec{X} + \theta_Y \vec{Y} + \theta_Z \vec{Z})$$

Ces relations écrites sous forme de matrice deviennent

$$\underbrace{\begin{bmatrix} u \\ v \\ w \\ \beta_x \\ \beta_y \end{bmatrix}}_{\underline{U}_{2d}} = \underbrace{\begin{bmatrix} \vec{x} \cdot \vec{X} & \vec{x} \cdot \vec{Y} & \vec{x} \cdot \vec{Z} & 0 & 0 & 0 \\ \vec{y} \cdot \vec{X} & \vec{y} \cdot \vec{Y} & \vec{y} \cdot \vec{Z} & 0 & 0 & 0 \\ \vec{z} \cdot \vec{X} & \vec{z} \cdot \vec{Y} & \vec{z} \cdot \vec{Z} & 0 & 0 & 0 \\ 0 & 0 & 0 & \vec{y} \cdot \vec{X} & \vec{y} \cdot \vec{Y} & \vec{y} \cdot \vec{Z} \\ 0 & 0 & 0 & -\vec{x} \cdot \vec{X} & -\vec{x} \cdot \vec{Y} & -\vec{x} \cdot \vec{Z} \end{bmatrix}}_{\mathbf{T}} \underbrace{\begin{bmatrix} U_X \\ U_Y \\ U_Z \\ \theta_X \\ \theta_Y \\ \theta_Z \end{bmatrix}}_{\underline{U}_{3d}}$$

où la matrice \mathbf{T} de taille 5×6 est appelée matrice de transformation. L'énergie de déformation vaut alors

$$\mathcal{E}_D = \frac{1}{2} \underline{U}_{2d}^T \mathbf{K}_{2d}^e \underline{U}_{2d} = \frac{1}{2} \underline{U}_{3d}^T \underbrace{\mathbf{T}^T \mathbf{K}_{2d}^e \mathbf{T}}_{=\mathbf{K}_{3d}^e} \underline{U}_{3d}$$

où \mathbf{K}_{2d}^e est la matrice de rigidité de l'élément de plaque dans son plan vue dans la section précédente.

La matrice de rigidité de l'élément plaque dans l'espace vaut donc

$$\mathbf{K}_{3d}^e = \mathbf{T}^T \mathbf{K}_{2d}^e \mathbf{T}.$$

XII — Décomposition de domaine

XII.1 Objectifs

Les objectifs de la décomposition de domaine sont multiples :

- utiliser un ordinateur à architecture parallèle, ou plusieurs ordinateurs connectés en réseau (cluster),
- coupler des structures constituant un système mécanique,
- coupler des formulations différentes (ex : couplage fluide-structure)

L'idée est de séparer le "gros" problème initial en plusieurs "petits" problèmes à résoudre séparément.

La méthode de Schur⁽¹⁾ primale porte son nom du fait que les inconnues nodales de l'interface séparant les domaines sont des variables de déplacements, c'est à dire les mêmes que celles des sous-domaines. La méthode de Schur duale introduit pour sa part la variable duale du déplacement à l'interface, soit le vecteur contrainte.

XII.2 Méthode de Schur primale

XII.2.1 Principe

Un problème classique à résoudre en statique linéaire par la méthode des éléments finis abouti à la résolution du système

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

(1) mathématicien russe, 1875-1941

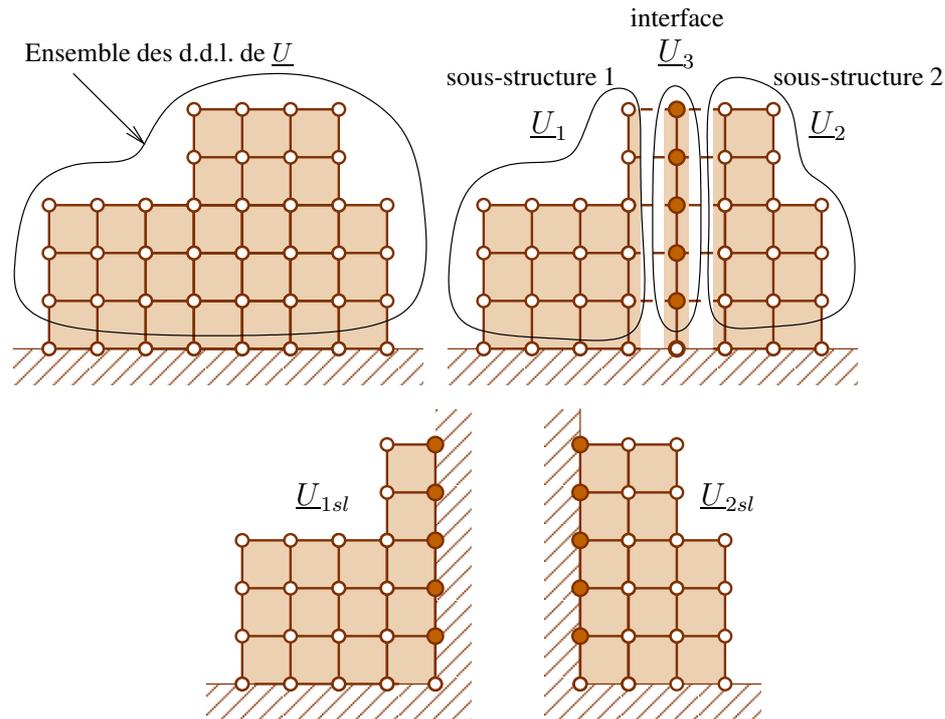


Figure XII.1 – Décomposition en deux sous-structures.

où \mathbf{K} est la matrice de rigidité, \underline{U} est le vecteur des inconnues de déplacements nodaux et \underline{F} est le vecteur des forces nodales. Afin de simplifier les notations, le vecteur \underline{U} ne contient que les degrés de liberté non bloqués. Le problème mécanique associé est représenté sur la figure XII.1.

Par soucis de simplicité, la structure est décomposée en seulement deux sous-structures (numérotées 1 et 2). Elles ne sont pas connectées entre elles mais elles sont en contact avec l'interface (numérotée 3). L'ensemble des nœuds de la structure est renuméroté de façon à avoir les inconnues nodales appartenant à la sous-structure 1 dans le vecteur \underline{U}_1 . On fait de même pour la sous-structure 2 et l'interface 3 pour les vecteurs \underline{U}_2 et \underline{U}_3 . En utilisant ces notations, le système à résoudre devient

$$\begin{bmatrix} \mathbf{K}_1 & \mathbf{0} & \mathbf{K}_{13} \\ \mathbf{0} & \mathbf{K}_2 & \mathbf{K}_{23} \\ \mathbf{K}_{13}^T & \mathbf{K}_{23}^T & \mathbf{K}_3 \end{bmatrix} \begin{bmatrix} \underline{U}_1 \\ \underline{U}_2 \\ \underline{U}_3 \end{bmatrix} = \begin{bmatrix} \underline{F}_1 \\ \underline{F}_2 \\ \underline{F}_3 \end{bmatrix}$$

On remarque que dans ce système, il n'y a pas de couplage entre les sous-structures puisqu'elles sont choisies de façon à être déconnectées.

Exercice XII.1 Proposer une numérotation des nœuds du maillage de la figure XII.1 afin d'obtenir la séparation entre les sous-domaines 1, 2 et 3. ■

La première étape est la résolution du problème sans liaison pour chaque sous-structure. Cela revient à résoudre indépendamment les deux systèmes suivants :

$$\mathbf{K}_1 \underline{U}_{1sl} = \underline{F}_1, \text{ et } \mathbf{K}_2 \underline{U}_{2sl} = \underline{F}_2.$$

Les deux vecteurs \underline{U}_{1sl} et \underline{U}_{2sl} obtenus sont complétés par les solutions avec liaison en introdui-

sant les vecteurs \underline{U}_{1al} et \underline{U}_{2al} tels que

$$\underline{U} = \underline{U}_{1sl} + \underline{U}_{1al}, \quad \text{et} \quad \underline{U} = \underline{U}_{2sl} + \underline{U}_{2al}.$$

On remarque que cette première étape dite "sans liaison" revient à résoudre pour chaque sous-structure un problème pour lequel l'interface est encastrée (Fig. XII.1).

En remplaçant dans le système initial, il vient

$$\begin{bmatrix} \mathbf{K}_1 & \mathbf{0} & \mathbf{K}_{13} \\ \mathbf{0} & \mathbf{K}_2 & \mathbf{K}_{23} \\ \mathbf{K}_{13}^T & \mathbf{K}_{23}^T & \mathbf{K}_3 \end{bmatrix} \begin{bmatrix} \underline{U}_{1sl} + \underline{U}_{1al} \\ \underline{U}_{2sl} + \underline{U}_{2al} \\ \underline{U}_3 \end{bmatrix} = \begin{bmatrix} \underline{F}_1 \\ \underline{F}_2 \\ \underline{F}_3 \end{bmatrix}.$$

Les deux premiers systèmes donnent

$$\mathbf{K}_1 \underline{U}_{1sl} + \mathbf{K}_1 \underline{U}_{1al} + \mathbf{K}_{13} \underline{U}_3 = \underline{F}_1,$$

et

$$\mathbf{K}_2 \underline{U}_{2sl} + \mathbf{K}_2 \underline{U}_{2al} + \mathbf{K}_{23} \underline{U}_3 = \underline{F}_2.$$

En remarquant que $\mathbf{K}_1 \underline{U}_{1sl} = \underline{F}_1$ et $\mathbf{K}_2 \underline{U}_{2sl} = \underline{F}_2$, il vient

$$\mathbf{K}_1 \underline{U}_{1al} + \mathbf{K}_{13} \underline{U}_3 = \mathbf{0}, \quad \text{et} \quad \mathbf{K}_2 \underline{U}_{2al} + \mathbf{K}_{23} \underline{U}_3 = \mathbf{0}.$$

Le troisième système d'équations donne

$$\mathbf{K}_{13}^T \underline{U}_{1sl} + \mathbf{K}_{13}^T \underline{U}_{1al} + \mathbf{K}_{23}^T \underline{U}_{2sl} + \mathbf{K}_{23}^T \underline{U}_{2al} + \mathbf{K}_3 \underline{U}_3 = \underline{F}_3$$

soit

$$\mathbf{K}_{13}^T \underline{U}_{1al} + \mathbf{K}_{23}^T \underline{U}_{2al} + \mathbf{K}_3 \underline{U}_3 = \underline{F}_3 - \mathbf{K}_{13}^T \underline{U}_{1sl} - \mathbf{K}_{23}^T \underline{U}_{2sl}$$

En notant

$$\underline{B} = \underline{F}_3 - \mathbf{K}_{13}^T \underline{U}_{1sl} - \mathbf{K}_{23}^T \underline{U}_{2sl}$$

on a alors

$$\mathbf{K}_{13}^T \underline{U}_{1al} + \mathbf{K}_{23}^T \underline{U}_{2al} + \mathbf{K}_3 \underline{U}_3 = \underline{B}$$

Finalement, le système à résoudre pour trouver \underline{U}_{1al} , \underline{U}_{2al} et \underline{U}_3 est

$$\begin{bmatrix} \mathbf{K}_1 & \mathbf{0} & \mathbf{K}_{13} \\ \mathbf{0} & \mathbf{K}_2 & \mathbf{K}_{23} \\ \mathbf{K}_{13}^T & \mathbf{K}_{23}^T & \mathbf{K}_3 \end{bmatrix} \begin{bmatrix} \underline{U}_{1al} \\ \underline{U}_{2al} \\ \underline{U}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \underline{B} \end{bmatrix}.$$

Les deux premières lignes du système donnent

$$\underline{U}_{1al} = -\mathbf{K}_1^{-1} \mathbf{K}_{13} \underline{U}_3 \quad \text{et} \quad \underline{U}_{2al} = -\mathbf{K}_2^{-1} \mathbf{K}_{23} \underline{U}_3$$

soit en remplaçant dans la dernière ligne

$$-\mathbf{K}_{13}^T \mathbf{K}_1^{-1} \mathbf{K}_{13} \underline{U}_3 - \mathbf{K}_{23}^T \mathbf{K}_2^{-1} \mathbf{K}_{23} \underline{U}_3 + \mathbf{K}_3 \underline{U}_3 = \underline{B}$$

En notant

$$\mathbf{S} = \mathbf{K}_3 - \mathbf{K}_{13}^T \mathbf{K}_1^{-1} \mathbf{K}_{13} - \mathbf{K}_{23}^T \mathbf{K}_2^{-1} \mathbf{K}_{23}$$

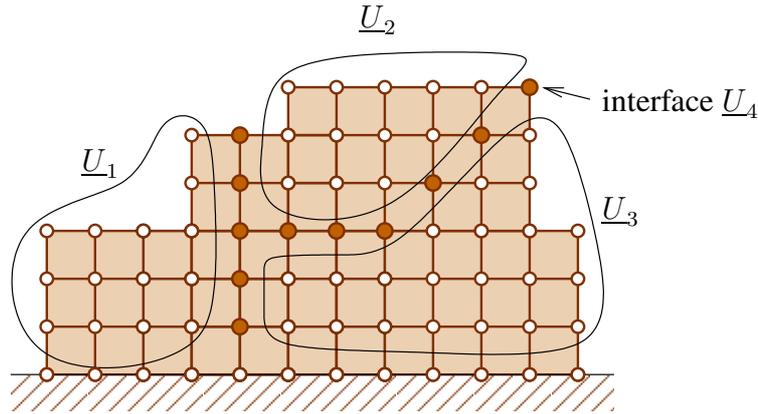


Figure XII.2 – Décomposition en $n - 1$ sous-structures avec $n = 4$.

on a alors le système suivant à résoudre :

$$\mathbf{S}\underline{U}_3 = \underline{B}$$

La matrice \mathbf{S} est appelée le complément de Schur.

Une fois que la solution \underline{U}_3 est calculée, il suffit de remplacer dans les expressions de \underline{U}_{1al} et \underline{U}_{2al} écrites en fonction de \underline{U}_3 pour avoir toutes les inconnues.

XII.2.2 Stratégie de résolution globale

Les différentes étapes sont donc :

1. Résolution sur 2 processeurs des problèmes sans liaison :

$$\mathbf{K}_1 \underline{U}_{1sl} = \underline{F}_1, \text{ et } \mathbf{K}_2 \underline{U}_{2sl} = \underline{F}_2.$$

2. Calcul du complément de Schur :

$$\mathbf{S} = \mathbf{K}_3 - \mathbf{K}_{13}^T \mathbf{K}_1^{-1} \mathbf{K}_{13} - \mathbf{K}_{23}^T \mathbf{K}_2^{-1} \mathbf{K}_{23}$$

3. Résolution de l'interface :

$$\mathbf{S}\underline{U}_3 = \underline{F}_3 - \mathbf{K}_{13}^T \underline{U}_{1sl} - \mathbf{K}_{23}^T \underline{U}_{2sl}$$

4. Calcul du problème avec liaison

$$\underline{U}_{1al} = -\mathbf{K}_1^{-1} \mathbf{K}_{13} \underline{U}_3, \text{ et } \underline{U}_{2al} = -\mathbf{K}_2^{-1} \mathbf{K}_{23} \underline{U}_3.$$

5. Construction de la solution complète :

$$\underline{U} = \underline{U}_{1sl} + \underline{U}_{1al}, \text{ et } \underline{U} = \underline{U}_{2sl} + \underline{U}_{2al}.$$

XII.2.3 Généralisation à plusieurs sous-structures

L'extension à plusieurs sous-structures se fait de façon naturelle. Avec un découpage en $n-1$ sous-structures (Fig. XII.2) et en notant n l'interface entre les sous-structures, on a alors l'algorithme suivant :

1. Résolution sur $n - 1$ processeurs des problèmes sans liaison :

$$\mathbf{K}_i \underline{U}_{isl} = \underline{F}_i,$$

pour $i = 1$ à $n - 1$

2. Calcul du complément de Schur :

$$\mathbf{S} = \mathbf{K}_n - \sum_{i=1}^{n-1} \mathbf{K}_{in}^T \mathbf{K}_i^{-1} \mathbf{K}_{in}$$

3. Résolution de l'interface :

$$\mathbf{S} \underline{U}_n = \underline{F}_n - \sum_{i=1}^{n-1} \mathbf{K}_{in}^T \underline{U}_{isl}$$

4. Calcul du problème avec liaison

$$\underline{U}_{ial} = -\mathbf{K}_i^{-1} \mathbf{K}_{in} \underline{U}_n$$

pour $i = 1$ à $n - 1$

5. Construction de la solution complète :

$$\underline{U}_i = \underline{U}_{isl} + \underline{U}_{ial}$$

pour $i = 1$ à $n - 1$

Exercice XII.2 Reprendre la démarche de résolution pour 3 sous-structures et une interface comme indiqué sur la figure XII.2 ■

XII.2.4 Calcul du complément de Schur

Le complément de Schur dans le cas de deux sous-structures s'écrit

$$\mathbf{S} = \mathbf{K}_3 - \mathbf{K}_{13}^T \mathbf{K}_1^{-1} \mathbf{K}_{13} - \mathbf{K}_{23}^T \mathbf{K}_2^{-1} \mathbf{K}_{23}.$$

Ce calcul fait intervenir l'inverse des matrices de rigidité \mathbf{K}_1 et \mathbf{K}_2 . Ces matrices étant en général très importantes, il n'est pas réaliste d'en calculer directement l'inverse. De plus \mathbf{K}_1^{-1} est multipliée à droite par \mathbf{K}_{13} et à gauche par \mathbf{K}_{13}^T ce qui donne finalement une matrice dont la taille est réduite à celle de l'interface en général très faible.

Une façon de calculer directement le produit $\mathbf{K}_1^{-1} \mathbf{K}_{13}$ sans calculer l'inverse de \mathbf{K}_1 est de résoudre une succession de problèmes en imposant les déplacements des nœuds de l'interface à 0 ou 1 (Fig. XII.3) : la i ème colonne de $\mathbf{K}_1^{-1} \mathbf{K}_{13}$ est exactement la solution \underline{X}_i du problème suivant :

$$\begin{bmatrix} \mathbf{K}_1 & \mathbf{K}_{13} \\ \mathbf{K}_{13}^T & \mathbf{K}_3 \end{bmatrix} \begin{bmatrix} \underline{X}_i \\ \underline{I}_i \end{bmatrix} = \begin{bmatrix} \underline{0} \\ \underline{Y}_i \end{bmatrix}$$

où \underline{I}_i est le déplacement unitaire imposé à l'interface défini par

$$\underline{I}_i = [0 \dots 0 \ 1 \ 0 \dots 0]^T \quad (\text{le } 1 \text{ est en } i\text{ème ligne})$$

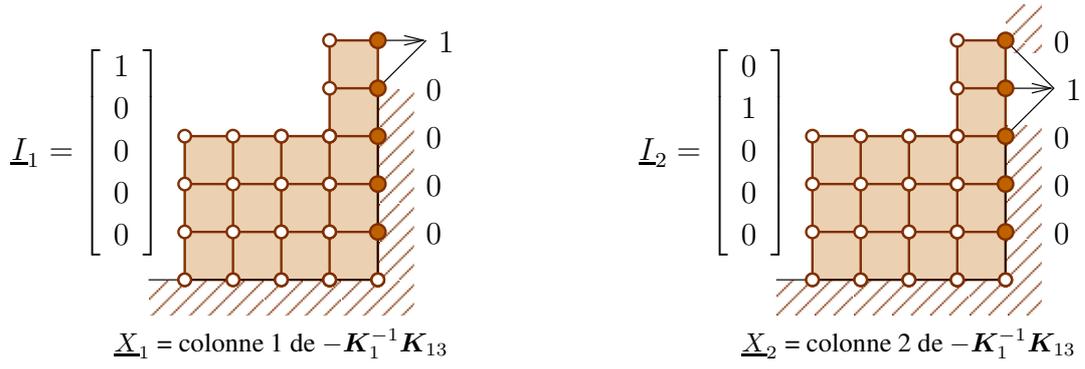


Figure XII.3 – Déplacement unitaire imposé à l’interface pour le calcul du complément de Schur.

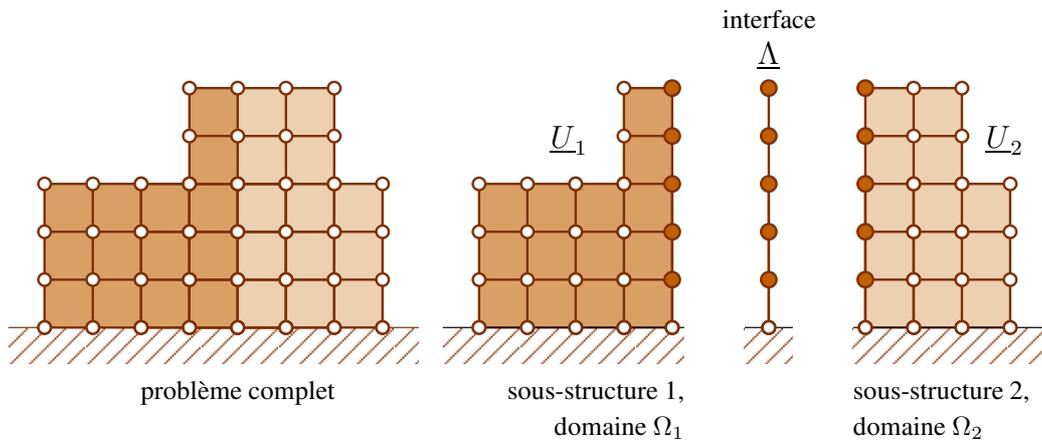


Figure XII.4 – Décomposition en 2 sous-structures par méthode Schur duale.

et où \underline{Y}_i sont les efforts de réaction inconnus du problème associé. En effet, le premier groupe d’équations s’écrit

$$\mathbf{K}_1 \underline{X}_i + \mathbf{K}_{13} \underline{L}_i = 0$$

ce qui donne la solution

$$\underline{X}_i = -\mathbf{K}_1^{-1} \mathbf{K}_{13} \underline{L}_i$$

soit la i ème colonne de $-\mathbf{K}_1^{-1} \mathbf{K}_{13}$.

En notant m le nombre de degré de liberté de l’interface, il suffit de résoudre successivement le problème avec

$$\underline{L}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \underline{L}_2 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \quad \dots \quad \underline{L}_m = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

pour avoir toutes les m colonnes de $-\mathbf{K}_1^{-1} \mathbf{K}_{13}$.

XII.3 Méthode de Schur duale

XII.3.1 Principe

La méthode de Schur duale fait intervenir les inconnues d'efforts à l'interface. Par soucis de simplicité on fait le développement pour deux sous-structures, leurs domaines sont notés Ω_1 et Ω_2 (Fig. XII.4) et on néglige les forces volumiques. Les nœuds de l'interface sont cette fois compris dans chacune des sous-structures. Ils sont aussi compris dans le maillage de l'interface.

On ajoute aux équations locales des domaines Ω_1 et Ω_2 la condition de continuité du déplacement à l'interface Γ

$$\vec{u}_1 = \vec{u}_2 \quad \text{sur } \Gamma$$

ainsi que la continuité du vecteur contrainte à l'interface

$$\sigma_1 \vec{n}_1 = -\sigma_2 \vec{n}_2 \quad \text{sur } \Gamma$$

où \vec{n}_1 et \vec{n}_2 sont respectivement les normales unitaires sortantes de Ω_1 et Ω_2 . En notant $\vec{\lambda}$ le vecteur contrainte à l'interface on a

$$\sigma_1 \vec{n}_1 = \vec{\lambda} \quad \text{sur } \Gamma \quad \text{et} \quad \sigma_2 \vec{n}_2 = -\vec{\lambda} \quad \text{sur } \Gamma$$

L'idée est d'ajouter ce vecteur contrainte aux inconnues du problème. Avec cette convention, $\vec{\lambda}$ est l'action de Ω_2 sur Ω_1 à travers l'interface Γ . L'interface est une frontière des domaines Ω_1 et Ω_2 sur laquelle les efforts extérieurs valent respectivement $\vec{\lambda}$ et $-\vec{\lambda}$. Le vecteur contrainte $\vec{\lambda}$ est souvent appelé *multiplicateur de Lagrange*.

La formulation variationnelle dans le domaine Ω_1 est alors :

Trouver \vec{u}_1 cinématiquement admissible, tel que $\forall \delta \vec{u}_1$ avec $\delta \vec{u}_1 = \vec{0}$ sur $\partial_u \Omega_1$ on ait

$$\int_{\Omega_1} \delta \epsilon_1^T \mathbf{C}_1 \epsilon_1 \, d\Omega - \int_{\partial_F \Omega_1} \delta u_1^T \vec{F}_1^d \, dS - \int_{\Gamma} \delta u_1^T \vec{\lambda} \, dS = 0$$

La formulation variationnelle dans le domaine Ω_2 est :

Trouver \vec{u}_2 cinématiquement admissible, tel que $\forall \delta \vec{u}_2$ avec $\delta \vec{u}_2 = \vec{0}$ sur $\partial_u \Omega_2$ on ait

$$\int_{\Omega_2} \delta \epsilon_2^T \mathbf{C}_2 \epsilon_2 \, d\Omega - \int_{\partial_F \Omega_2} \delta u_2^T \vec{F}_2^d \, dS + \int_{\Gamma} \delta u_2^T \vec{\lambda} \, dS = 0$$

Enfin, la condition de continuité de déplacement est assurée en multipliant l'équation $\vec{u}_1 = \vec{u}_2$ par le vecteur contrainte virtuel $\delta \vec{\lambda}$ et en intégrant le long de Γ , ce qui s'écrit

$$\int_{\Gamma} \delta \vec{\lambda}^T \vec{u}_1 \, dS - \int_{\Gamma} \delta \vec{\lambda}^T \vec{u}_2 \, dS = 0$$

XII.3.2 Discrétisation

La discrétisation dans les domaines Ω_1 et Ω_2 se fait de façon classique :

$$\vec{u}_1 = \Phi_1 \underline{U}_1 \quad \text{et} \quad \vec{u}_2 = \Phi_2 \underline{U}_2$$

où Φ_1 et Φ_2 sont les matrices des fonctions de forme des sous-structures 1 et 2 et \underline{U}_1 et \underline{U}_2 sont les déplacements nodaux associés. Le long de l'interface le vecteur contrainte est discrétisé par

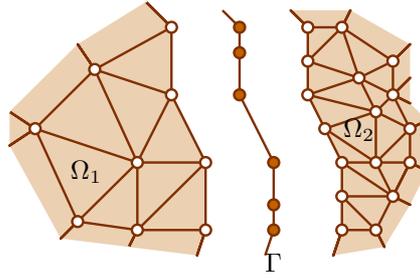


Figure XII.5 – Couplage de deux maillages incompatibles.

des fonctions de forme. En tridimensionnel et en supposant que l'interface contient m nœuds, on peut écrire

$$\vec{\lambda} = \begin{bmatrix} \lambda_x \\ \lambda_y \\ \lambda_z \end{bmatrix} = \begin{bmatrix} \phi_1 & \dots & \phi_m & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & \phi_1 & \dots & \phi_m & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & \phi_1 & \dots & \phi_m \end{bmatrix} \underline{\Lambda} = \Phi_\lambda \underline{\Lambda}$$

avec $\underline{\Lambda} = [\lambda_{x1} \dots \lambda_{xm} \lambda_{y1} \dots \lambda_{ym} \lambda_{z1} \dots \lambda_{zm}]^T$.

En remplaçant dans la formulation variationnelle, on trouve de façon standard les matrices de rigidité \mathbf{K}_1 et \mathbf{K}_2 des deux sous-structures, les efforts extérieurs \underline{F}_1 et \underline{F}_2 sur les deux sous-structures ainsi que les matrices de couplages $\mathbf{C}_{1\lambda}$ et $\mathbf{C}_{2\lambda}$ telles que

$$\int_{\Gamma} \vec{\delta u}_1^T \vec{\lambda} dS = \delta \underline{U}_1^T \int_{\Gamma} \Phi_1^T \Phi_\lambda dS \underline{\Lambda} = -\delta \underline{U}_1^T \mathbf{C}_{1\lambda} \underline{\Lambda}$$

$$\int_{\Gamma} \vec{\delta u}_2^T \vec{\lambda} dS = \delta \underline{U}_2^T \int_{\Gamma} \Phi_2^T \Phi_\lambda dS \underline{\Lambda} = \delta \underline{U}_2^T \mathbf{C}_{2\lambda} \underline{\Lambda}$$

Finalement, la formulation variationnelle discrétisée s'écrit :

$$\delta \underline{U}_1^T \mathbf{K}_1 \underline{U}_1 + \delta \underline{U}_1^T \mathbf{C}_{1\lambda} \underline{\Lambda} = \delta \underline{U}_1^T \underline{F}_1 ; \quad \delta \underline{U}_2^T \mathbf{K}_2 \underline{U}_2 + \delta \underline{U}_2^T \mathbf{C}_{2\lambda} \underline{\Lambda} = \delta \underline{U}_2^T \underline{F}_2 ; \quad \delta \underline{\Lambda}^T \mathbf{C}_{1\lambda}^T \underline{U}_1 + \delta \underline{\Lambda}^T \mathbf{C}_{2\lambda}^T \underline{U}_2 = 0$$

ce qui donne le système suivant à résoudre :

$$\begin{bmatrix} \mathbf{K}_1 & \mathbf{0} & \mathbf{C}_{1\lambda} \\ \mathbf{0} & \mathbf{K}_2 & \mathbf{C}_{2\lambda} \\ \mathbf{C}_{1\lambda}^T & \mathbf{C}_{2\lambda}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{U}_1 \\ \underline{U}_2 \\ \underline{\Lambda} \end{bmatrix} = \begin{bmatrix} \underline{F}_1 \\ \underline{F}_2 \\ \underline{0} \end{bmatrix}$$

Ce système peut être résolu en suivant la démarche de la méthode de Schur primale (avec $\mathbf{K}_3 = \mathbf{0}$).

XII.3.3 Couplage de maillages incompatibles

L'avantage dans cette seconde approche est qu'il n'y a *a priori* pas besoin que les maillages des sous-structures soient compatibles à l'interface (Fig. XII.5). Ceci permet donc de raccorder des maillages incompatibles provenant par exemples de bureaux d'études différents maillant les pièces d'un même assemblage.

Le choix du maillage de l'interface dans ce cas peut être effectué de différentes façons, ceci n'est pas détaillé dans ce cours.

XIII — Initiation à la programmation

XIII.1 Langage de programmation

Un langage de programmation est une interface entre l'homme et l'ordinateur qui permet de donner des successions d'ordres à celui-ci. Les ordres sont écrits dans un fichier texte sans mise en forme, appelé fichier source, à l'aide d'un éditeur de texte (à ne pas confondre avec un traitement de texte). Ce fichier source est ensuite lu, puis transformé en langage machine par l'ordinateur qui peut alors l'exécuter.

Les langages ont beaucoup évolué depuis le premier langage de programmation (assembleur) qui était très difficile à comprendre. Les langages modernes permettent d'utiliser des fonctions (ou sous-routines), et sont, pour certains, orientés objet.

Les langages peuvent être interprétés ou compilés :

- langage interprété : les instructions du programme sont lues, interprétées en langage machine puis exécutées au fur et à mesure. L'avantage réside dans la facilité de lecture et de modification du programme, l'inconvénient est la lenteur d'exécution.
- langage compilé : le programme est lu puis transformé en langage machine par un compilateur pour créer un fichier exécutable. Ce fichier est ensuite exécuté. Les avantages sont d'abord d'avoir une vitesse d'exécution du programme très importante, ensuite de pouvoir donner le fichier exécutable à une personne tierce afin qu'il puisse l'utiliser sans lui dévoiler le fichier source. L'inconvénient est que après chaque modification du fichier source, il faut le re-compiler, ce qui rend la programmation plus lourde.

```
# librairie de calcul scientifique
import scipy

print("Premier_programme_en_Python")

a=3.14
b=2.78
i=3
w=1+1.0j

print("a_est_un_reel:",a)
print("b_est_un_reel:",b)
print("i_est_un_entier:",i)
print("w_est_un_complexe:",w)

c=a*b
print("a*b=: ",c)

c=scipy.sqrt(a)
print("racine_carree_de_a=: ",c)

i=i+1
print("on_ajoute_1_a_i:",i)
i=i+1
print("on_rajoute_encore_1_a_i:",i)
```

Figure XIII.1 – Premier programme en Python

XIII.2 Le langage Python

XIII.2.1 Les bases du langage Python et son installation

Le langage Python est un langage interprété, orienté objet. Des bibliothèques de calcul scientifique sont utilisables. Ces bibliothèques sont compilées, ce qui rend malgré tout le langage performant pour le calcul par éléments finis.

Un premier programme en Python (version 3.4) est proposé sur la figure XIII.1. L'indentation des programmes est très importante, dans cet exemple qui est une suite d'actions sans boucles ni conditions, toutes les lignes commencent à la première colonne. Tout ce qui suit le caractère # n'est pas interprété par Python, c'est du commentaire. La bibliothèque de calcul scientifique `scipy` est d'abord chargée en mémoire. On écrit un message à l'écran, puis on affecte ensuite aux variables `a` et `b` des valeurs réelles (type `float`). L'affectation mémoire et la déclaration du type de variable sont automatiquement effectuées par Python, ce qui rend la programmation plus facile mais qui ralentit l'exécution en contrepartie. On affecte à la variable `i` la valeur entière 3 (type `int`). Si on lui avait affecté 3.0 alors la variable `i` aurait été de type `float`. On affecte à la variable `w` la valeur complexe $1 + j$ (type `complex`). On affiche à l'écran les contenus de ces 3 variables, puis on effectue quelques calculs en appelant la fonction racine carrée `sqrt` de la bibliothèque `scipy`. Enfin, on incrémente `i` de 1 deux fois de suite. Le signe `=` dans un programme est interprété comme une

```

Premier programme en Python
a est un reel : 3.14
b est un reel : 2.78
i est un entier : 3
w est un complexe : (1+1j)
a*b = : 8.7292
racine carree de a = : 1.77200451467
on ajoute 1 a i : 4
on rajoute encore 1 a i : 5

```

Figure XIII.2 – Résultat de l’affichage à l’écran du premier programme en Python

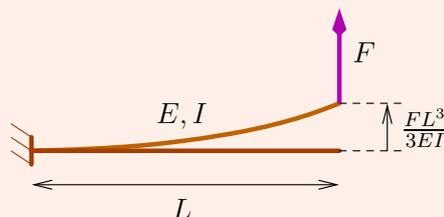
affectation de variable, c’est à dire que l’instruction $i=i+1$ ajoute 1 à i : ce n’est pas une équation à une inconnue à résoudre. Le résultat de l’affichage de ce programme est donné sur la figure XIII.2.

L’installation de Python est relativement simple :

- Sous Ubuntu : via synaptic, installer : `python3.4`, `idle-python3.4`, `python-scipy`, `python-numpy` et peut-être d’autres paquets si besoin...
- Sous Windows et Mac OS : installer la suite Anaconda qui contient Python, les librairies de calcul scientifique et un éditeur de texte intégré.

Pour programmer, le plus pratique est d’utiliser un IDE (en anglais pour “integrated development environment”, c’est à dire environnement de développement intégré) : par exemple `idle-python` mais il en existe d’autres. Une fenêtre d’exécution s’ouvre, on peut éditer les fichiers en les ouvrant dans un éditeur de texte intégré (`file/open`) puis les exécuter avec `run`.

Exercice XIII.1 Écrire un programme en Python qui calcule la flèche d’une poutre encastree-libre à partir de sa longueur L , de son module d’Young E , de son moment quadratique de section I et de l’effort appliqué F .



XIII.2.2 Les conditions et boucles en Python

Le programme présenté sur la figure XIII.3 est un jeu de devinette très simple. On cache un chiffre i (ici 6) au joueur qu’il doit deviner. A chaque essai on lui dit si sa proposition est plus grande ou plus petite que le chiffre caché. Il a droit à 3 essais. Ce programme fait appel à une boucle conditionnée par la valeur de la variable j qui compte le nombre d’essais. On remarque qu’il y a un premier niveau d’indentation qui correspond aux instructions de la boucle `while`, puis un deuxième niveau qui correspond aux tests de conditions `if`. Il n’y a pas de `end` en Python, la fin d’une boucle ou d’une condition est marquée par la disparition de l’indentation. Cette obligation d’indentation rend les programmes écrits en Python très lisibles.

```

print("Devinez_mon_chiffre ,vous_avez_3_essais.")

i=6 # Chiffre a deviner
j=1 # Compteur des essais

while (j <= 3):
    print("Essai_",j)
    print("Entrez_votre_proposition:")
    k = int(input())

    if (k==i):
        print("Bravo")
        j=99

    if (k<i):
        print("Mon_chiffre_est_plus_grand")

    if (k>i):
        print("Mon_chiffre_est_plus_petit")

    j = j+1

if (j==4):
    print("Vous_n'avez_pas_trouve ,mon_chiffre_etait_",i)

```

Figure XIII.3 – Jeu de devinette en Python

```

import random

print("Revision_de_la_table_de_multiplication.")
k=0 # Compteur des bons resultats

for i in range(20):
    a=int(random.uniform(2,10))
    b=int(random.uniform(2,10))
    print(a,"*",b,"=")
    c = int(input('??'))
    if c==(a*b):
        print("Oui")
        k=k+1
    else :
        print("Non")

print("Note_sur_20:",k)

```

Figure XIII.4 – Révision de la table de multiplication en Python

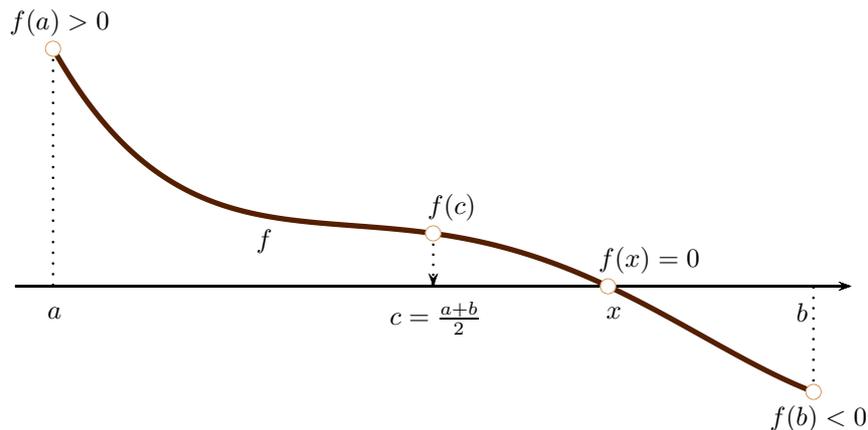


Figure XIII.5 – Recherche du zéro d'une fonction par dichotomie

Le programme de la figure XIII.4 permet de réviser la table de multiplication et d'afficher une note sur 20. Grâce à la librairie `random`, on choisit deux entiers au hasard entre 1 et 10 : la fonction `int` transforme le réel tiré au hasard entre 2 et 10 avec `random.uniform` en un entier. On affiche ces 2 entiers `a` et `b` à l'écran. On demande à l'utilisateur le résultat de la multiplication que l'on stocke dans la variable `c`, puis on compare à la bonne réponse. Si la réponse est bonne, on affiche oui et on incrémente le compteur `k` des bonnes réponses, sinon on affiche non. Ce compteur doit être initialisé à zéro au début du programme. On affiche enfin la note sur 20.

Exercice XIII.2 Écrire un programme en Python qui calcule la suite de Syracuse à partir d'un entier donné par l'utilisateur. Le principe est de partir d'un nombre entier plus grand que zéro ; s'il est pair, on le divise par 2 ; s'il est impair, on le multiplie par 3 et on ajoute 1. En répétant l'opération, on obtient une suite d'entiers positifs dont chacun ne dépend que de son prédécesseur. La conjecture de Syracuse est que cette suite calculée depuis n'importe quel entier strictement positif atteint 1. On peut utiliser la commande python `x%y` qui renvoie le reste de la division de x par y . ■

XIII.2.3 Les fonctions en Python

Une fonction possède un nom, des entrées, des sorties, des instructions et des variables propres, et fait des appels éventuels à d'autres fonctions. Afin d'illustrer le principe de fonction en Python, on écrit un programme qui calcule le zéro d'une fonction à une variable par dichotomie. L'algorithme est le suivant (voir illustration sur la figure XIII.5) :

Initialisation de a et b

Tant que $b - a > \text{tolérance}$ **faire**

$$c = \frac{a+b}{2}$$

Si $f(a) \times f(c) > 0$ **alors** $a \leftarrow c$ **sinon** $b \leftarrow c$

Afficher c

Afficher le nombre d'itérations

Ce programme est donné sur la figure XIII.6. La fonction mathématique pour laquelle on souhaite connaître le zéro est entrée dans la fonction Python que l'on nomme `mafonction`. L'entrée est `x`, la sortie est `valeur`. Ici, on a choisit de chercher le zéro de la fonction cosinus. Le

```

def mafonction(x):
    import scipy
    value = scipy.cos(x)
    return value

print("Zero_de_la_fonction_cosinus_entre_0_et_3.")
a=0.0
b=3.0
j=1 # Compteur des iterations
while ((abs(a-b))>1e-8):
    j=j+1
    ya=mafonction(a)
    c=0.5*(a+b)
    yc=mafonction(c)
    if ((ya*yc)>0.0):
        a=c
    else:
        b=c

print("nombre_d'iterations :",j)
print("Zero_de_la_fonction:",c)

```

Figure XIII.6 – Calcul du zéro par dichotomie en Python

programme principal applique l'algorithme de recherche de zéro par dichotomie, il appelle la fonction `mafonction` plusieurs fois. Pour changer de fonction mathématique, il suffit de changer son expression dans `mafonction`.

Exercice XIII.3 Programmer en Python le calcul de l'intégrale d'une fonction $f(x)$, par exemple $\cos x$, entre a et b par la formule des rectangles avec n intervalles à définir. ■

XIII.2.4 Le calcul matriciel en Python

La librairie `scipy` permet de faire du calcul matriciel. Pour les petites matrices comme les matrices de rigidité élémentaires (jusqu'à 100×100), elles sont stockées complètement en utilisant le type `array`. La figure XIII.7 présente un exemple de multiplication de matrices. Les matrices sont écrites ligne par ligne. La fonction `dot` permet de faire une multiplication de matrices, toutes les autres opérations classiques sur les matrices sont possibles.

Lorsque les matrices sont de grandes tailles, comme les matrices de rigidité de structures (jusqu'à $1\,000\,000 \times 1\,000\,000$), elles sont stockées comme des matrices creuses. Le principe est de stocker chaque terme non nul en lui associant son emplacement en ligne et colonne dans la matrice. Par exemple, on souhaite stocker au format creux la matrice suivante :

$$\begin{bmatrix} 4.6 & 6.9 & 0 & 0 & 0 \\ 1.5 & 1 & 3.8 & 9.2 & 0 \\ 0 & 3.0 & 2 & 0 & 0 \\ 0 & 9 & 0 & 5.5 & 7 \\ 0 & 0 & 0 & 7 & 8 \end{bmatrix}$$

```

import scipy

A=scipy.array([[1.1 , 0.0],
               [ 8.7 , 0.0],
               [ 0.0 , 2.6]])

B=scipy.array([[6.1 , 4.5 , 0.0 , 0.0],
               [0.0 , 5.5 , 3.7 , 1.9]])

C=scipy.dot(A,B)

print C

```

Figure XIII.7 – Déclaration de matrices et calculs matriciels en Python

```

import scipy
import scipy.sparse

V=scipy.array([4.6,1.0,2.0,5.5,8.0,6.9,1.5
               ,3.8,3.0,9.2,9.0,7.0,7.0])
I=scipy.array([0,1,2,3,4,0,1,1,2,1,3,3,4])
J=scipy.array([0,1,2,3,4,1,0,2,1,3,1,4,3])

A=scipy.sparse.csc_matrix( (V,(I,J)), shape=(5,5) )

print A.todense()

```

Figure XIII.8 – Déclaration de matrices au format creux

On stocke les valeurs des termes de la matrice dans le tableau V de réels, les indices des ligne et colonne correspondant à chaque terme sont stockés dans deux tableaux de pointeurs I et J . Les pointeurs en Python commencent à 0, c'est à dire que la première ligne d'une matrice est la ligne 0. On peut commencer par stocker les termes de la diagonale, puis les autres ensuite, mais l'ordre n'a pas d'importance :

$$\begin{aligned}
 V &= [4.6 \ 1 \ 2 \ 5.5 \ 8 \ 6.9 \ 1.5 \ 3.8 \ 3.0 \ 9.2 \ 9 \ 7 \ 7] \\
 I &= [0 \ 1 \ 2 \ 3 \ 4 \ 0 \ 1 \ 1 \ 2 \ 1 \ 3 \ 3 \ 4] \\
 J &= [0 \ 1 \ 2 \ 3 \ 4 \ 1 \ 0 \ 2 \ 1 \ 3 \ 1 \ 4 \ 3]
 \end{aligned}$$

Le programme Python correspondant à cet exemple est donné sur la figure XIII.8.

L'avantage du stockage creux des matrices est que l'on peut répéter plusieurs fois les mêmes indices de lignes et colonnes avec des valeurs différentes, les valeurs s'additionnent automatiquement. Par exemple, on veut assembler les deux matrices suivantes 2×2 dans une matrice 3×3 :

$$\begin{bmatrix} 10 & -10 \\ -10 & 10 \end{bmatrix}$$

```

import scipy
import scipy.sparse

V=scipy.array([10,-10,-10,10,20,-20,-20,20])
I=scipy.array([0,0,1,1,1,1,2,2])
J=scipy.array([0,1,0,1,1,2,1,2])

A=scipy.sparse.csc_matrix( (V,(I,J)), shape=(3,3) )

print A.todense()

```

Figure XIII.9 – Assemblage de matrices au format creux

sur les lignes et colonnes [0,1], et

$$\begin{bmatrix} 20 & -20 \\ -20 & 20 \end{bmatrix}$$

sur les lignes et colonnes [1,2]. Le résultat attendu est :

$$\begin{bmatrix} 10 & -10 & 0 \\ -10 & 10+20 & -20 \\ 0 & -20 & 20 \end{bmatrix} = \begin{bmatrix} 10 & -10 & 0 \\ -10 & 30 & -20 \\ 0 & -20 & 20 \end{bmatrix}$$

Le stockage peut se faire avec les 3 tableaux suivants :

$$\begin{aligned} V &= [10 & -10 & -10 & 10 & 20 & -20 & -20 & 20] \\ I &= [0 & 0 & 1 & 1 & 1 & 1 & 2 & 2] \\ J &= [0 & 1 & 0 & 1 & 1 & 2 & 1 & 2] \end{aligned}$$

où 10 et 20 sont stockés séparément en [1,1]. Le programme Python correspondant est donné en figure XIII.9.

Exercice XIII.4 Écrire un programme qui assemble automatiquement la matrice

$$\begin{bmatrix} 10 & -10 \\ -10 & 10 \end{bmatrix}$$

dans une matrice 5×5 en assemblant sur les lignes et colonnes [0,1],[1,2],[2,3],[0,3],[3,4]. On pourra avantageusement utiliser un tableau de connectivité `element` tel que

```
elements=scipy.array([[0,1],[1,2],[2,3],[0,3],[3,4]]).
```

XIII.3 Le langage Fortran

XIII.3.1 Les bases du langage Fortran et son installation

Le langage Fortran est un langage compilé. C'est un langage relativement ancien qui est utilisé depuis longtemps par les scientifiques : la majorité des codes de calcul par éléments finis est écrit en Fortran.

Un exemple de programme écrit en Fortran est donné sur la figure XIII.10. Les lignes du programme écrit en Fortran commencent à la 7ème colonne, elles ne doivent pas dépasser la

```

PROGRAM Bonjour
C commentaires: Declaration des variables
  real a,b,c
  integer i
  complex w

C   Corps du programme principal
C   En fortran , on commence en 7eme colonne
c   on arrete en 72eme:
C23456-----|
  print*, 'Programme_qui_dit_bonjour'

  a=3.14
  b=2.78
  i=3
  w=cplx(1,1)

  print*, 'a_est_un_reel:',a
  print*, 'b_est_un_reel:',b
  print*, 'i_est_un_entier:',i
  print*, 'w_est_un_complexe:',w

  c=a*b
  print*, 'a*b=:',c
  c=sqrt(a)
  print*, 'racine_carree_de_a=:',c

  i=i+1
  print*, 'on_ajoute_1_a_i:',i
  i=i+1
  print*, 'on_rajoute_encore_1_a_i:',i
END

```

Figure XIII.10 – Premier programme en Fortran

72ème colonne. Ce programme exécute les mêmes actions que son équivalent en Python de la figure XIII.1. La première ligne du programme commence par la commande PROGRAM En langage Fortran, il faut déclarer les variables avant de les utiliser, c'est à dire préciser leurs types et leurs tailles. Ici, a et b sont des réels déclarés de type real ; la variable i est un entier déclaré de type integer.

L'installation de Fortran est relativement simple :

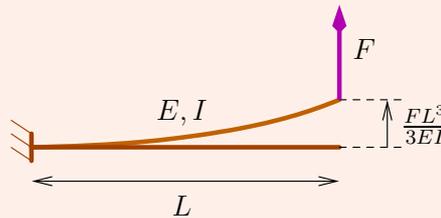
- Sous Ubuntu : via synaptic, installer : gfortran
- Sous Windows : télécharger gfortran depuis le site <http://gcc.gnu.org/wiki/GFortranBinaries#Windows>, puis suivre les instructions.

Pour exécuter le programme, il faut le compiler pour en faire un programme exécutable. Il suffit de taper la commande suivante dans un terminal (en supposant que le fichier s'appelle `bonjour.f`) :

```
gfortran bonjour.f
```

Cela a pour effet de créer un fichier exécutable `a.out` que l'on peut lancer en tapant `a.out` dans le terminal.

Exercice XIII.5 Écrire un programme en Fortran qui calcule la flèche d'une poutre encastree-libre à partir de sa longueur L , de son module d'Young E , de son moment quadratique de section I et de l'effort appliqué F .



XIII.3.2 Les conditions et boucles en Fortran

L'exemple de la figure XIII.11 reprend l'exemple du jeu de devinette précédent afin de montrer les équivalences des boucles et conditions en Fortran.

L'exemple de la figure XIII.12 reprend l'exemple du jeu de révision de la table de multiplication.

Exercice XIII.6 Écrire un programme en Fortran qui calcule la suite de Syracuse à partir d'un entier donné par l'utilisateur. Le principe est de partir d'un nombre entier plus grand que zéro ; s'il est pair, on le divise par 2 ; s'il est impair, on le multiplie par 3 et on ajoute 1. En répétant l'opération, on obtient une suite d'entiers positifs dont chacun ne dépend que de son prédécesseur. La conjecture de Syracuse est que cette suite calculée depuis n'importe quel entier strictement positif atteint 1.

XIII.3.3 Les fonctions en Fortran

L'exemple de la figure XIII.13 reprend l'exemple de recherche du zéro d'une fonction par dichotomie afin de montrer les équivalences des déclarations de fonctions en Fortran.

```
PROGRAM Jeu
integer i,j,k

print*, '- Devinez mon chiffre ,vous avez 3 essais.'
i=4  ! Chiffre a deviner
j=1  ! Compteur des essais
do while (j.le.3)
  print*, 'Essai_',j
  print*, 'Entrez votre proposition:'
  read*,k

  if (k.eq.i) then
    print*, 'Bravo'
    return ! Termine le programme
  endif

  if (k.lt.i) then
    print*, 'Mon chiffre est plus grand'
  endif

  if (k.gt.i) then
    print*, 'Mon chiffre est plus petit'
  endif

  j = j+1
enddo

print*, 'Vous n\'avez pas trouve ,mon chiffre etait_',i
END
```

Figure XIII.11 – Jeu de devinette en Fortran

```
PROGRAM Jeu
implicit none
integer i,k,a,b,c
real rand

print*, '- Revision de la table de multiplication .'
k=0 ! Compteur des bons resultats
do i=1,20
  a=floor(8*rand(0))+2
  b=floor(8*rand(0))+2
  print*, a, '*',b, '=?'
  read*,c
  if (c.eq.(a*b)) then
    print*, 'Oui'
    print*, ' '
    k=k+1
  else
    print*, 'Non'
    print*, ' '
  endif
enddo
print*, 'Note sur 20: ',k
END
```

Figure XIII.12 – Révision de la table de multiplication en Fortran

```
PROGRAM Dichotomie
integer i,j,k
double precision a,b,c,ya,yc,mafonction
print*, 'Zero_de_la_fonction_cosinus_entre_0_et_3.'
a=0.0
b=3.0
j=1 ! Compteur des iterations
do while ((abs(a-b)).ge.1e-6)
    j=j+1
    ya=mafonction(a)
    c=0.5*(a+b)
    yc=mafonction(c)
    if ((ya*yc).ge.(0.0)) then
        a=c
    else
        b=c
    endif
enddo
print*, 'nombre_d''iterations:',j
print*, 'Zero_de_la_fonction:',c
END

double precision function mafonction(x)
    double precision x
    mafonction = cos(x)
RETURN
END
```

Figure XIII.13 – Calcul du zéro par dichotomie en Fortran

```

PROGRAM Produit de matrices

  double precision A(3,2),B(2,4),C(3,4)
  integer i,j,k

  A(1,1)=1.1 ; A(1,2)=0.0
  A(2,1)=8.7 ; A(2,2)=0.0
  A(3,1)=0.0 ; A(3,2)=2.6

  B(1,1)=6.1 ; B(1,2)=4.5 ; B(1,3)=0.0 ; B(1,4)=0.0
  B(2,1)=0.0 ; B(2,2)=5.5 ; B(2,3)=3.7 ; B(2,4)=1.9

  do i=1,3
    do j=1,4
      C(i,j)=0.0
      do k=1,2
        C(i,j)=C(i,j)+A(i,k)*B(k,j)
      enddo
    enddo
  enddo

  do i=1,3
    print*, (C(i,j), j=1,4)
  enddo

END

```

Figure XIII.14 – Déclaration de matrices et calculs matriciels en Fortran

Exercice XIII.7 Programmer en Fortran le calcul de l'intégrale d'une fonction $f(x)$, par exemple $\cos x$, entre a et b par la formule des rectangles avec n intervalles à définir. ■

XIII.3.4 Le calcul matriciel en Fortran

Bien qu'il existe des bibliothèques de calcul matriciel en Fortran, on peut programmer les multiplications de matrices en imbriquant 3 boucles comme dans l'exemple de la figure XIII.14. Le langage Fortran ne permet pas à priori d'utiliser des matrices au format creux (sparse).

XIII.4 Utilisation de routines Fortran dans un programme en Python

On peut facilement utiliser des routines écrites en Fortran dans un programme écrit en Python. On peut alors combiner les avantages des deux langages : vitesse d'exécution du Fortran (notamment pour les boucles) et souplesse de programmation de Python (notamment pour la manipulation des matrices sparses).

Par exemple, on peut écrire en Fortran une routine qui calcule le produit de deux matrices. Cette routine, donnée sur la figure XIII.15, est sauvegardée sous le nom `lib_multiplication.f`. On construit une bibliothèque Python à partir de cette routine Fortran en

```

SUBROUTINE prodmat(A,B,C,l,m,n)
  integer l,m,n,i,j,k
  real*8 A(l,m),B(m,n),C(l,n)

C entrees/sorties pour Python
Cf2py intent(in) A
Cf2py intent(in) B
Cf2py intent(out) C

  do i=1,l
    do j=1,n
      C(i,j)=0.0
      do k=1,m
        C(i,j)=C(i,j)+A(i,k)*B(k,j)
      enddo
    enddo
  enddo
return
END

```

Figure XIII.15 – Routine Fortran de calcul de produit de deux matrices

tapant dans le terminal :

```
f2py3 -c -m lib_multiplication lib_multiplication.f
```

Les commandes suivantes (qui sont des commentaires pour Fortran) dans la routine prodmat :

```
Cf2py intent(in) A
```

```
Cf2py intent(in) B
```

```
Cf2py intent(out) C
```

sont interprétées par f2py afin de construire les entrées et les sorties de la fonction prodmat de la librairie Python ainsi créée lib_multiplication.cpython-34m.so. Cette librairie peut alors être utilisée dans un programme Python comme dans l'exemple de la figure XIII.16. Dans cet exemple, les tailles des matrices paramétrées par les entiers l, m et n sont automatiquement gérées par f2py et n'ont pas besoin d'être précisées lors de l'appel de la routine depuis le programme Python. La commande Python suivante :

```
print(lib_multiplication.__doc__)
```

permet de connaître toutes les fonctions de la librairie lib_multiplication. La commande Python suivante :

```
print(lib_multiplication.prodmat.__doc__)
```

permet de connaître les entrées et les sorties attendues de la fonction prodmat.

Le résultat de l'affichage du programme Python de la figure XIII.16 qui fait appel à une librairie Fortran est donné sur la figure XIII.17.

```

import scipy
import lib_multiplication

A=scipy.array([[1.1 , 0.0],
               [ 8.7 , 0.0],
               [ 0.0 , 2.6]])

B=scipy.array([[6.1 , 4.5 , 0.0 , 0.0],
               [0.0 , 5.5 , 3.7 , 1.9]])

print(lib_multiplication.__doc__)
print(lib_multiplication.prodmat.__doc__)

C=lib_multiplication.prodmat(A,B)

print(C)

```

Figure XIII.16 – Utilisation d'une librairie écrite en Fortran dans un programme en Python

This module 'lib_multiplication' is auto-generated with f2py (version:2).

Functions:

```
c = prodmat(a,b,l=shape(a,0),m=shape(a,1),n=shape(b,1))
```

```
c = prodmat(a,b,[l,m,n])
```

Wrapper for 'prodmat'.

Parameters

a : input rank-2 array('d') with bounds (l,m)

b : input rank-2 array('d') with bounds (m,n)

Other Parameters

l : input int, optional

Default: shape(a,0)

m : input int, optional

Default: shape(a,1)

n : input int, optional

Default: shape(b,1)

Returns

c : rank-2 array('d') with bounds (l,n)

```

[[ 6.71  4.95  0.    0. ]
 [ 53.07 39.15  0.    0. ]
 [  0.   14.3  9.62  4.94]]

```

Figure XIII.17 – Résultat de l'affichage de l'utilisation d'une librairie écrite en Fortran dans un programme en Python

XIV — Code SILEX



XIV.1 Présentation

SILEX est un code de calcul par éléments finis écrit en Python et en Fortran. La figure XIV.1 montre les articulations entre les langages ainsi que les liens avec le logiciel Gmsh.

SILEX est articulé de la façon suivante :

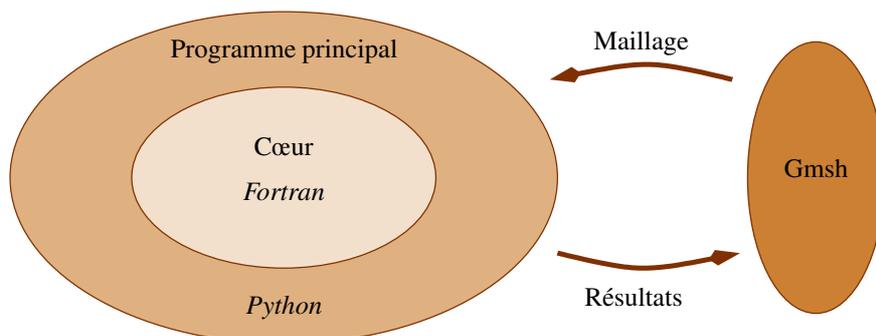


Figure XIV.1 – Articulation du code SILEX

- Le langage Python est utilisé pour :
 - définir les paramètres du calcul,
 - lire le maillage,
 - résoudre le système,
 - écrire les résultats.
- Le langage Fortran est utilisé pour effectuer les calculs au niveau local et construire la matrice de rigidité. Le langage python peut aussi être utilisé pour cette partie, l'avantage étant que SILEX fonctionne alors entièrement en Python, l'inconvénient est que les temps de calculs sont plus longs.
- Le logiciel libre Gmsh est utilisé pour faire les maillages et visualiser les résultats.

Le lien suivant donne une partie du code :

<http://antoinelegay.free.fr/silex/silex.html>.

Seules les routines pour le tétraèdre à 4 nœuds dans le cas de statique linéaire sont disponibles pour l'instant en téléchargement libre. Un premier exemple est traité sous forme de tutoriel sur ce même lien.

XIV.2 Cœur écrit en Fortran de SILEX pour le tétraèdre à 4 nœuds

XIV.2.1 Matrice de rigidité élémentaire

On rappelle que les fonctions de forme de l'un élément tétraèdre à 4 nœuds sont solutions du système d'équations suivant :

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ x_a & x_b & x_c & x_d \\ y_a & y_b & y_c & y_d \\ z_a & z_b & z_c & z_d \end{bmatrix} \begin{bmatrix} \varphi_a(x, y, z) \\ \varphi_b(x, y, z) \\ \varphi_c(x, y, z) \\ \varphi_d(x, y, z) \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix}$$

En résolvant le système par la méthode de Cramer, on obtient les expressions suivantes :

$$\varphi_i(x, y, z) = \frac{1}{6v}(\alpha_i + \beta_i x + \gamma_i y + \delta_i z)$$

avec

$$\alpha_i = (-1)^{p+1} \begin{vmatrix} x_j & x_k & x_l \\ y_j & y_k & y_l \\ z_j & z_k & z_l \end{vmatrix} \quad \beta_i = (-1)^p \begin{vmatrix} 1 & 1 & 1 \\ y_j & y_k & y_l \\ z_j & z_k & z_l \end{vmatrix}$$

$$\gamma_i = (-1)^{p+1} \begin{vmatrix} 1 & 1 & 1 \\ x_j & x_k & x_l \\ z_j & z_k & z_l \end{vmatrix} \quad \delta_i = (-1)^p \begin{vmatrix} 1 & 1 & 1 \\ x_j & x_k & x_l \\ y_j & y_k & y_l \end{vmatrix}$$

où v est le volume de l'élément et en utilisant la permutation donnée dans le tableau suivant :

i	j	k	l	p	i	j	k	l	p	i	j	k	l	p	i	j	k	l	p
a	b	c	d	1	b	a	c	d	0	c	a	b	d	1	d	a	b	c	0

Le volume v de l'élément est le déterminant de la matrice suivante :

$$v = \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{vmatrix}$$

L'opérateur gradient discrétisé de l'élément tétraèdre à 4 nœuds s'écrit :

$$\mathbf{B}^e = \frac{1}{6v} \begin{bmatrix} \beta_a & 0 & 0 & \beta_b & 0 & 0 & \beta_c & 0 & 0 & \beta_d & 0 & 0 \\ 0 & \gamma_a & 0 & 0 & \gamma_b & 0 & 0 & \gamma_c & 0 & 0 & \gamma_d & 0 \\ 0 & 0 & \delta_a & 0 & 0 & \delta_b & 0 & 0 & \delta_c & 0 & 0 & \delta_d \\ 0 & \delta_a & \gamma_a & 0 & \delta_b & \gamma_b & 0 & \delta_c & \gamma_c & 0 & \delta_d & \gamma_d \\ \delta_a & 0 & \beta_a & \delta_b & 0 & \beta_b & \delta_c & 0 & \beta_c & \delta_d & 0 & \beta_d \\ \gamma_a & \beta_a & 0 & \gamma_b & \beta_b & 0 & \gamma_c & \beta_c & 0 & \gamma_d & \beta_d & 0 \end{bmatrix}$$

L'expression de la matrice de rigidité élémentaire est

$$\mathbf{K}^e = \int_{\Omega_e} \mathbf{B}^{eT} \mathbf{C} \mathbf{B}^e d\Omega_e$$

Étant donné que \mathbf{B}^e est constant, cette expression devient

$$\mathbf{K}^e = v \mathbf{B}^{eT} \mathbf{C} \mathbf{B}^e$$

La loi de comportement \mathbf{C} vaut

$$\mathbf{C} = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix}$$

Le calcul de v fait intervenir le déterminant d'une matrice 4×4 avec des 1 sur la première ligne, on calcule de façon analytique ce déterminant puis on en crée une fonction Fortran `det44_ligne_de_un`.

Les calculs des coefficients β_i , γ_i et δ_i font intervenir le déterminant d'une matrice 3×3 avec des 1 sur la première ligne, on calcule de façon analytique ce déterminant puis on en crée une fonction Fortran `det33_ligne_de_un`.

XIV.2.2 Matrice de rigidité globale

La routine `StiffnessMatrix` effectue une boucle sur les éléments tétraèdres à 4 nœuds afin de calculer leurs matrices de rigidité et de les assembler. Les entrées de cette routine sont :

- les coordonnées des nœuds dans le tableau `nodes` de taille `nb. nœuds × 3`,
- la table de connectivité dans le tableau `elements` de taille `nb. éléments × 4`,
- les propriétés matériau dans le tableau `material` (module d'Young E et coefficient de Poisson ν).

La sortie de la routine est la matrice de rigidité globale sous forme sparse dans les 2 tableaux I_k et J_k de pointeurs et le tableau V_k contenant les valeurs des matrices de rigidité élémentaires.

On calcule dans un premier temps la loi de comportement du matériau C en fonction de E et ν . Dans un second temps, pour chaque élément e , on extrait les coordonnées des nœuds puis on calcule la matrice de rigidité élémentaire k_e de taille 12×12 .

La routine `StiffnessMatrix` écrite en Python est aussi donnée dans la suite.


```

double precision beta(4), gamm(4), delt(4), det44_ligne_de_un
double precision det33_ligne_de_un, BB(6,12)

young = material(1)
nu    = material(2)
do i=1,6
  do j=1,6
    CC(i, j)=0.0d0
  enddo
  do j=1,12
    BB(i, j)=0.0d0
  enddo
enddo
lambda = nu*young/((1+nu)*(1-2*nu))
mu     = young/(2*(1+nu))
do i=1,3
  CC(i, i)=lambda+2*mu
  CC(i+3, i+3)=mu
enddo
CC(1,2)=lambda;CC(2,1)=lambda
CC(1,3)=lambda;CC(3,1)=lambda
CC(2,3)=lambda;CC(3,2)=lambda

p=1
do e=1, nbelem

  do i=1,4
    idnodes(i) = elements(e, i)
    python indexing
    dofx(i)    = (idnodes(i)-1)*3
    dofy(i)    = (idnodes(i)-1)*3+1
    dofz(i)    = (idnodes(i)-1)*3+2
  enddo
  do i=1,4
    dofelem(1+3*(i-1)) = dofx(i)
    dofelem(2+3*(i-1)) = dofy(i)
    dofelem(3+3*(i-1)) = dofz(i)
  enddo

  do i=1,4
    X(i)=nodes(idnodes(i),1)
    Y(i)=nodes(idnodes(i),2)
    Z(i)=nodes(idnodes(i),3)
  enddo

  a34(1,1)=X(1); a34(1,2)=X(2); a34(1,3)=X(3); a34(1,4)=X(4)
  a34(2,1)=Y(1); a34(2,2)=Y(2); a34(2,3)=Y(3); a34(2,4)=Y(4)
  a34(3,1)=Z(1); a34(3,2)=Z(2); a34(3,3)=Z(3); a34(3,4)=Z(4)
  det_of_sys = det44_ligne_de_un(a34)
  Vol        = abs(det_of_sys/6)

  a23(1,1) = Y(2); a23(1,2) = Y(3); a23(1,3) = Y(4)
  a23(2,1) = Z(2); a23(2,2) = Z(3); a23(2,3) = Z(4)
  beta(1)=-det33_ligne_de_un(a23)

  a23(1,1) = X(2); a23(1,2) = X(3); a23(1,3) = X(4)
  a23(2,1) = Z(2); a23(2,2) = Z(3); a23(2,3) = Z(4)

```

```

gamm(1)=+ det33_ligne_de_un(a23)

a23(1,1) = X(2); a23(1,2) = X(3); a23(1,3) = X(4)
a23(2,1) = Y(2); a23(2,2) = Y(3); a23(2,3) = Y(4)
delt(1)=- det33_ligne_de_un(a23)

a23(1,1) = Y(1); a23(1,2) = Y(3); a23(1,3) = Y(4)
a23(2,1) = Z(1); a23(2,2) = Z(3); a23(2,3) = Z(4)
beta(2)=+ det33_ligne_de_un(a23)

a23(1,1) = X(1); a23(1,2) = X(3); a23(1,3) = X(4)
a23(2,1) = Z(1); a23(2,2) = Z(3); a23(2,3) = Z(4)
gamm(2)=- det33_ligne_de_un(a23)

a23(1,1) = X(1); a23(1,2) = X(3); a23(1,3) = X(4)
a23(2,1) = Y(1); a23(2,2) = Y(3); a23(2,3) = Y(4)
delt(2)=+ det33_ligne_de_un(a23)

a23(1,1) = Y(1); a23(1,2) = Y(2); a23(1,3) = Y(4)
a23(2,1) = Z(1); a23(2,2) = Z(2); a23(2,3) = Z(4)
beta(3)=- det33_ligne_de_un(a23)

a23(1,1) = X(1); a23(1,2) = X(2); a23(1,3) = X(4)
a23(2,1) = Z(1); a23(2,2) = Z(2); a23(2,3) = Z(4)
gamm(3)=+ det33_ligne_de_un(a23)

a23(1,1) = X(1); a23(1,2) = X(2); a23(1,3) = X(4)
a23(2,1) = Y(1); a23(2,2) = Y(2); a23(2,3) = Y(4)
delt(3)=- det33_ligne_de_un(a23)

a23(1,1) = Y(1); a23(1,2) = Y(2); a23(1,3) = Y(3)
a23(2,1) = Z(1); a23(2,2) = Z(2); a23(2,3) = Z(3)
beta(4)=+ det33_ligne_de_un(a23)

a23(1,1) = X(1); a23(1,2) = X(2); a23(1,3) = X(3)
a23(2,1) = Z(1); a23(2,2) = Z(2); a23(2,3) = Z(3)
gamm(4)=- det33_ligne_de_un(a23)

a23(1,1) = X(1); a23(1,2) = X(2); a23(1,3) = X(3)
a23(2,1) = Y(1); a23(2,2) = Y(2); a23(2,3) = Y(3)
delt(4)=+ det33_ligne_de_un(a23)

do j=1,4
  BB(1,1+3*(j-1)) = beta(j)/(6.0 d0*Vol)
  BB(2,2+3*(j-1)) = gamm(j)/(6.0 d0*Vol)
  BB(3,3+3*(j-1)) = delt(j)/(6.0 d0*Vol)
  BB(4,2+3*(j-1)) = delt(j)/(6.0 d0*Vol)
  BB(4,3+3*(j-1)) = gamm(j)/(6.0 d0*Vol)
  BB(5,1+3*(j-1)) = delt(j)/(6.0 d0*Vol)
  BB(5,3+3*(j-1)) = beta(j)/(6.0 d0*Vol)
  BB(6,1+3*(j-1)) = gamm(j)/(6.0 d0*Vol)
  BB(6,2+3*(j-1)) = beta(j)/(6.0 d0*Vol)
enddo

ke = matmul(transpose(BB), matmul(CC, BB))*Vol

do i=1,12

```

```

do j=1,12
  Ik(p)=dofelem(i)
  Jk(p)=dofelem(j)
  Vk(p)=ke(i,j)
  p=p+1
enddo
enddo

enddo
return
end

```

Routine Python du cœur de SILEX permettant de calculer la matrice de rigidité :

```

def stiffnessmatrix(nodes, elements, material):

    nelelem = elements.shape[0]
    Ik = scipy.zeros(12*12*nelelem, dtype=int)
    Jk = scipy.zeros(12*12*nelelem, dtype=int)
    Vk = scipy.zeros(12*12*nelelem, dtype=float)
    beta = scipy.zeros((4), dtype=float)
    gamm = scipy.zeros((4), dtype=float)
    delt = scipy.zeros((4), dtype=float)
    A34 = scipy.zeros((3,4), dtype=float)
    A23 = scipy.zeros((3,4), dtype=float)
    B = scipy.zeros((6,12), dtype=float)
    dofelem = scipy.zeros(12, dtype=int)
    dofx = scipy.zeros(4, dtype=int)
    dofy = scipy.zeros(4, dtype=int)
    dofz = scipy.zeros(4, dtype=int)
    idnodes = scipy.zeros(4, dtype=int)
    young = material[0]
    nu = material[1]
    lamb = nu*young/((1+nu)*(1-2*nu))
    mu = young/(2*(1+nu))
    C = scipy.array([[lamb+2*mu, lamb, lamb, 0.0, 0.0, 0.0],
                    [lamb, lamb+2*mu, lamb, 0.0, 0.0, 0.0],
                    [lamb, lamb, lamb+2*mu, 0.0, 0.0, 0.0],
                    [0.0, 0.0, 0.0, mu, 0.0, 0.0],
                    [0.0, 0.0, 0.0, 0.0, mu, 0.0],
                    [0.0, 0.0, 0.0, 0.0, 0.0, mu]])

    p=0

    for e in range(nelelem):
        idnodes[:] = elements[e,:]-1
        dofx[:] = (idnodes)*3
        dofy[:] = (idnodes)*3+1
        dofz[:] = (idnodes)*3+2
        dofelem[0] = dofx[0]; dofelem[1] = dofy[0]; dofelem[2] = dofz[0]
        dofelem[3] = dofx[1]; dofelem[4] = dofy[1]; dofelem[5] = dofz[1]
        dofelem[6] = dofx[2]; dofelem[7] = dofy[2]; dofelem[8] = dofz[2]
        dofelem[9] = dofx[3]; dofelem[10] = dofy[3]; dofelem[11] = dofz[3]

        X=nodes[idnodes,0]; Y=nodes[idnodes,1]; Z=nodes[idnodes,2]
        A34[0,:] = X; A34[1,:] = Y; A34[2,:] = Z
        det_of_sys = det44_ligne_de_un(A34)
        Vol = abs(det_of_sys/6)

```

```

A23[0,0] = Y[1]; A23[0,1] = Y[2]; A23[0,2] = Y[3]
A23[1,0] = Z[1]; A23[1,1] = Z[2]; A23[1,2] = Z[3]
beta[0]= -det33_ligne_de_un(A23)
A23[0,0] = X[1]; A23[0,1] = X[2]; A23[0,2] = X[3]
gamm[0]= det33_ligne_de_un(A23)
A23[1,0] = Y[1]; A23[1,1] = Y[2]; A23[1,2] = Y[3]
delt[0]= -det33_ligne_de_un(A23)

```

```

A23[0,0] = Y[0]; A23[0,1] = Y[2]; A23[0,2] = Y[3]
A23[1,0] = Z[0]; A23[1,1] = Z[2]; A23[1,2] = Z[3]
beta[1]= det33_ligne_de_un(A23)
A23[0,0] = X[0]; A23[0,1] = X[2]; A23[0,2] = X[3]
gamm[1]= -det33_ligne_de_un(A23)
A23[1,0] = Y[0]; A23[1,1] = Y[2]; A23[1,2] = Y[3]
delt[1]= det33_ligne_de_un(A23)

```

```

A23[0,0] = Y[0]; A23[0,1] = Y[1]; A23[0,2] = Y[3]
A23[1,0] = Z[0]; A23[1,1] = Z[1]; A23[1,2] = Z[3]
beta[2]= -det33_ligne_de_un(A23)
A23[0,0] = X[0]; A23[0,1] = X[1]; A23[0,2] = X[3]
gamm[2]= det33_ligne_de_un(A23)
A23[1,0] = Y[0]; A23[1,1] = Y[1]; A23[1,2] = Y[3]
delt[2]= -det33_ligne_de_un(A23)

```

```

A23[0,0] = Y[0]; A23[0,1] = Y[1]; A23[0,2] = Y[2]
A23[1,0] = Z[0]; A23[1,1] = Z[1]; A23[1,2] = Z[2]
beta[3]= det33_ligne_de_un(A23)
A23[0,0] = X[0]; A23[0,1] = X[1]; A23[0,2] = X[2]
gamm[3]= -det33_ligne_de_un(A23)
A23[1,0] = Y[0]; A23[1,1] = Y[1]; A23[1,2] = Y[2]
delt[3]= det33_ligne_de_un(A23)

```

```

B[0,0] = beta[0]; B[0,3] = beta[1]; B[0,6] = beta[2]; B[0,9] = beta[3]
B[1,1] = gamm[0]; B[1,4] = gamm[1]; B[1,7] = gamm[2]; B[1,10] = gamm[3]
B[2,2] = delt[0]; B[2,5] = delt[1]; B[2,8] = delt[2]; B[2,11] = delt[3]
B[3,2] = gamm[0]; B[3,5] = gamm[1]; B[3,8] = gamm[2]; B[3,11] = gamm[3]
B[3,1] = delt[0]; B[3,4] = delt[1]; B[3,7] = delt[2]; B[3,10] = delt[3]
B[4,0] = delt[0]; B[4,3] = delt[1]; B[4,6] = delt[2]; B[4,9] = delt[3]
B[4,2] = beta[0]; B[4,5] = beta[1]; B[4,8] = beta[2]; B[4,11] = beta[3]
B[5,0] = gamm[0]; B[5,3] = gamm[1]; B[5,6] = gamm[2]; B[5,9] = gamm[3]
B[5,1] = beta[0]; B[5,4] = beta[1]; B[5,7] = beta[2]; B[5,10] = beta[3]

```

```

ke=scipy.dot(B.T, scipy.dot(C,B))*Vol/(det_of_sys*det_of_sys)

```

```

for i in range(12):
    Ik[p:(p+12)]=dofelem[i]
    Jk[p:(p+12)]=dofelem[:]
    Vk[p:(p+12)]=ke[i,:]
    p=p+12

```

```

return Ik, Jk, Vk

```

XIV.3 Programme principal en Python

Le listing du programme principal est dans la suite. Le programme principal écrit en Python pilote les calculs. Le détail des parties est le suivant :

— Partie *Import libraries* :

- Cette première partie indique à python les librairies spécifiques que l'on va utiliser.
- En déplaçant le # du commentaire, on peut choisir la librairie de calcul au niveau élémentaire :

— Pour une version Python (typiquement pour Windows) :

```
#import silex_lib_tet4_fortran as silex_lib_elt
import silex_lib_tet4_python as silex_lib_elt
```

— Pour une version avec une librairie Fortran compilée :

```
import silex_lib_tet4_fortran as silex_lib_elt
#import silex_lib_tet4_python as silex_lib_elt
```

— Partie *USER PART* : partie à modifier pour un nouveau calcul

— La variable `MeshFileName` indique le nom du fichier maillage provenant de gmsh (sans l'extension `.msh`).

— La variable `ResultsFileName` indique le nom du fichier dans lequel les résultats sont écrits (sans l'extension `.msh`).

— La variable `eltype` indique le type d'éléments que l'on utilise : 4 pour les tétraèdres à 4 nœuds. Les autres types d'éléments sont détaillés dans la librairie `silex_lib_gmsh.py`.

— La variable `ndim` indique la dimension de la géométrie : 2 pour 2D, 3 pour 3D.

— La variable `flag_write_fields` indique le type de résultats que l'on souhaite écrire dans le fichier résultat : 1 pour tous les champs, 0 pour une sortie courte. D'autres valeurs peuvent être définies à la fin du fichier `Main.py`.

— La routine `ReadGmshNodes` de la librairie `silex_lib_gmsh` permet d'importer les coordonnées des nœuds depuis le fichier de maillage dans le tableau `nodes`.

— Utilisation :

```
nodes=silex_lib_gmsh.ReadGmshNodes('monmaillage.msh', 2 ou 3)
```

— La routine `ReadGmshElements` de la librairie `silex_lib_gmsh` permet d'importer la table de connectivité des éléments depuis le fichier de maillage dans le tableau `elements`, les numéros des nœuds correspondants sont stockés dans le tableau `Idnodes`.

— Utilisation :

```
elements,Idnodes=silex_lib_gmsh.ReadGmshElements
('monmaillage.msh',element_type,physical_group)
```

— Une fois le maillage volume importé, on utilise la routine `ReadGmshElements` pour importer les maillages surfaciques (type d'élément 2 pour les triangles à 3 nœuds) des surfaces fonctionnelles du calcul : `elementsS1 ...` et `IdnodeS1 ...`

- Afin de vérifier si les maillages sont bien importés, on peut les écrire dans un fichier au format gmsh à l'aide de la routine `WriteResults` de la librairie `silex_lib_gmsh`.
 - Utilisation :


```
silex_lib_gmsh.WriteResults
('maillage_lu.msh', nodes, elements, element_type)
```
- La variable `Young` indique le module d'Young.
- La variable `nu` indique le coefficient de Poisson.
- Les nœuds bloqués dans la direction \vec{x} sont indiqués dans le tableau `IdNodesFixed_x`, ceux bloqués dans la direction \vec{y} dans le tableau `IdNodesFixed_y`, enfin ceux bloqués dans la direction \vec{z} dans le tableau `IdNodesFixed_z`.
- La routine `forceonsurface` de la librairie `silex_lib_tet4` (renommée en `silex_lib_elt`) permet de calculer les forces aux nœuds dues à une force surfacique de norme `press` dans la direction `direction` :
 - Utilisation :


```
F = silex_lib_elt.forceonsurface(nodes,
surfacic_elements_where_the_load_is_applied,press,direction)
```

Si la direction vaut `[0.0,0.0,0.0]`, alors c'est la normale locale à la surface qui est utilisée, c'est le cas pour une force surfacique due à une pression. Dans ce cas la direction de la normale étant calculée à partir de la numérotation locale des nœuds, elle peut être en sens opposé, il suffit de changer le signe de `press` pour obtenir une force dans la bonne direction.
- Partie *EXPERT* : partie inchangée lors d'un nouveau calcul
 - La variable `nnodes` indique le nombre de nœuds.
 - La variable `ndof` indique nombre de degrés de liberté.
 - La variable `nelem` indique nombre d'éléments.
 - La variable `Fixed_Dofs` contient les ddl bloqués.
 - Le tableau `SolvedDofs` indique les ddl libres à calculer.
 - Le tableau `Q` contient les déplacements des nœuds.
 - Calcul de la matrice de rigidité et assemblage au format sparse :


```
Ik,Jk,Vk=silex_lib_elt.stiffnessmatrix(nodes,elements,[Young,nu])
K=scipy.sparse.csc_matrix((Vk,(Ik,Jk)), shape=(ndof,ndof) ,dtype=float)
```
 - Résolution du système en déplacement (on suppose que les déplacements imposés sont nuls, à modifier dans le cas contraire) :


```
Q[SolvedDofs] = scipy.sparse.linalg.spsolve
(K[SolvedDofs,:][:,SolvedDofs],F[SolvedDofs])
```
 - Calcul des contraintes (dans les éléments et lissées aux nœuds), des déformations (dans les éléments et lissées aux nœuds), des contributions des éléments à l'erreur, et enfin de l'erreur en énergie `ZZ1` :

```
SigmaElem, SigmaNodes, EpsilonElem, EpsilonNodes, ErrorElem, ErrorGlobal  
= silex_lib_elt.compute_stress_strain_error(nodes,elements,[Young,nu],Q)
```

- Le tableau `disp` contient les déplacements des nœuds écrit sur 3 colonnes (ou 2 en 2D) pour la sortie au format `gmsh`.
- Le tableau `load` contient les efforts imposés aux nœuds écrit sur 3 colonnes (ou 2 en 2D) pour la sortie au format `gmsh`.
- La liste `fields_to_write` contient les champs à écrire dans le fichier résultat. La routine `WriteResults` de la librairie `silex_lib_gmsh` donne plus de détails.

```

# Python Main program
#####
# Import libraries
#####
import string; import time; import scipy;
import scipy.sparse; import scipy.sparse.linalg

import sys
sys.path.append('.././libraries')

## Choose between the Fortran or the Python librairie:

#import silex_lib_tet4_fortran as silex_lib_elt
import silex_lib_tet4_python as silex_lib_elt

import silex_lib_gmsh

#####
print("SILEX_CODE_-_calcul_avec_des_tet4")
#####

#####
# USER PART: Import mesh, boundary conditions and material
#####
# Input mesh: define the name of the mesh file (*.msh)
MeshFileName='MyMesh'

# Output result file: define the name of the result file (*.msh)
ResultsFileName='MyResults'

# choose the element type
eltype=4

# choose geometry dimension
ndim=3

# choose the results in the results file:
# 0 for short output
# 1 for long output
flag_write_fields=0

# read the mesh from gmsh
nodes=silex_lib_gmsh.ReadGmshNodes(MeshFileName+'.msh',ndim)
elements ,Idnodes=silex_lib_gmsh.ReadGmshElements(MeshFileName+'.msh',eltype,5)

# read surfaces where to impose boundary conditions
elementsS1 ,IdnodeS1=silex_lib_gmsh.ReadGmshElements(MeshFileName+'.msh',2,1)
elementsS2 ,IdnodeS2=silex_lib_gmsh.ReadGmshElements(MeshFileName+'.msh',2,2)
#...

# write the surface mesh in a gmsh-format file to verify if its correct
silex_lib_gmsh.WriteResults(ResultsFileName+'Volum',nodes,elements,4)
silex_lib_gmsh.WriteResults(ResultsFileName+'surf1',nodes,elementsS1,2)
#...

# Define material
Young = 200000.0

```

```

nu      = 0.3

# Boundary conditions , for instance :
IdNodesFixed_x=IdnodeS3
IdNodesFixed_y=IdnodeS1
IdNodesFixed_z=IdnodeS2

# compute external forces from pressure
press = 10 #MPa
# give the direction of the surfacic load :
#     if [0.0,0.0,0.0] then the local normal to the surface is used
#     otherwise , the direction is normalized to 1
direction = [0.0,0.0,0.0]
F = silex_lib_elt.forceonsurface (nodes , elementsS3 , press , direction )

#####
#     EXPERT PART
#####
#     initialisations
#####
# get number of nodes , dof and elements from the mesh
nnodes = nodes . shape [0]
ndof    = nnodes*ndim
nelem   = elements . shape [0]

# define fixed dof
Fixed_Dofs = scipy . hstack ( [(IdNodesFixed_x - 1)*3,
                              (IdNodesFixed_y - 1)*3+1,
                              (IdNodesFixed_z - 1)*3+2])

# define free dof
SolvedDofs = scipy . setdiff1d (range (ndof) , Fixed_Dofs)

# initialize displacement vector
Q=scipy . zeros (ndof)

#####
#     compute stiffness matrix
#####
Ik , Jk , Vk=silex_lib_elt . stiffnessmatrix (nodes , elements , [Young , nu ])
K=scipy . sparse . csc_matrix ( (Vk, (Ik , Jk)) , shape=(ndof , ndof) , dtype=float )

#####
#     Solve the problem
#####
Q[SolvedDofs] = scipy . sparse . linalg . spsolve (
                K[SolvedDofs , :][: , SolvedDofs] , F[SolvedDofs] )

#####
#     compute smooth stress and error in elements
#####
SigmaElem , SigmaNodes , EpsilonElem , EpsilonNodes , ErrorElem , ErrorGlobal=
    silex_lib_elt . compute_stress_strain_error (nodes , elements , [Young , nu ] , Q)

print ("The global error is : " , ErrorGlobal)

#####

```

```

#           Write results to gmsh format
#####
# displacement written on 3 columns:
disp=scipy.zeros((nnodes,ndim))
disp[range(nnodes),0]=Q[list(range(0,ndof,3))]
disp[range(nnodes),1]=Q[list(range(1,ndof,3))]
disp[range(nnodes),2]=Q[list(range(2,ndof,3))]

# external load written on 3 columns:
load=scipy.zeros((nnodes,ndim))
load[range(nnodes),0]=F[list(range(0,ndof,3))]
load[range(nnodes),1]=F[list(range(1,ndof,3))]
load[range(nnodes),2]=F[list(range(2,ndof,3))]

if flag_write_fields==0:
    fields_to_write=[ [ disp , 'nodal',ndim,'displacement' ],
                      [ SigmaElem[range(nelem),[6]], 'elemental',1,'Sigma_V.M.' ],
                      [ SigmaNodes[range(nnodes),[6]], 'nodal',1,'Smooth_Sigma_V.M.' ],
                      [ ErrorElem , 'elemental',1,'error' ],
                      ]

if flag_write_fields==1:
    fields_to_write=[ [ disp , 'nodal',ndim,'displacement' ],
                      [ load , 'nodal',ndim,'Force' ],
                      [ SigmaElem[range(nelem),[0]], 'elemental',1,'Sigma_11' ],
                      [ SigmaElem[range(nelem),[1]], 'elemental',1,'Sigma_22' ],
                      [ SigmaElem[range(nelem),[2]], 'elemental',1,'Sigma_33' ],
                      [ SigmaElem[range(nelem),[3]], 'elemental',1,'Sigma_23' ],
                      [ SigmaElem[range(nelem),[4]], 'elemental',1,'Sigma_13' ],
                      [ SigmaElem[range(nelem),[5]], 'elemental',1,'Sigma_12' ],
                      [ SigmaElem[range(nelem),[6]], 'elemental',1,'Sigma_V.M.' ],
                      [ SigmaNodes[range(nnodes),[0]], 'nodal',1,'Smooth_Sigma_11' ],
                      [ SigmaNodes[range(nnodes),[1]], 'nodal',1,'Smooth_Sigma_22' ],
                      [ SigmaNodes[range(nnodes),[2]], 'nodal',1,'Smooth_Sigma_33' ],
                      [ SigmaNodes[range(nnodes),[3]], 'nodal',1,'Smooth_Sigma_23' ],
                      [ SigmaNodes[range(nnodes),[4]], 'nodal',1,'Smooth_Sigma_13' ],
                      [ SigmaNodes[range(nnodes),[5]], 'nodal',1,'Smooth_Sigma_12' ],
                      [ SigmaNodes[range(nnodes),[6]], 'nodal',1,'Smooth_Sigma_V.M.' ],
                      [ EpsilonElem[range(nelem),[0]], 'elemental',1,'Epsilon_11' ],
                      [ EpsilonElem[range(nelem),[1]], 'elemental',1,'Epsilon_22' ],
                      [ EpsilonElem[range(nelem),[2]], 'elemental',1,'Epsilon_33' ],
                      [ EpsilonElem[range(nelem),[3]]/2.0, 'elemental',1,'Epsilon_23' ],
                      [ EpsilonElem[range(nelem),[4]]/2.0, 'elemental',1,'Epsilon_13' ],
                      [ EpsilonElem[range(nelem),[5]]/2.0, 'elemental',1,'Epsilon_12' ],
                      [ EpsilonNodes[range(nnodes),[0]], 'nodal',1,'Smooth_Epsilon_11' ],
                      [ EpsilonNodes[range(nnodes),[1]], 'nodal',1,'Smooth_Epsilon_22' ],
                      [ EpsilonNodes[range(nnodes),[2]], 'nodal',1,'Smooth_Epsilon_33' ],
                      [ EpsilonNodes[range(nnodes),[3]]/2.0, 'nodal',1,'Smooth_Epsilon_23' ],
                      [ EpsilonNodes[range(nnodes),[4]]/2.0, 'nodal',1,'Smooth_Epsilon_13' ],
                      [ EpsilonNodes[range(nnodes),[5]]/2.0, 'nodal',1,'Smooth_Epsilon_12' ],
                      [ ErrorElem , 'elemental',1,'error' ],
                      ]

# write the mesh and the results in a gmsh-format file
silex_lib_gmsh.WriteResults(ResultsFileName,nodes,elements,eltype,fields_to_write)
print("————_END_————")

```

XIV.4 Gmsh : un logiciel libre de maillage et de visualisation

Le logiciel de maillage et de visualisation utilisé par SILEX est Gmsh, c'est un logiciel libre et multi-plateforme. Ce logiciel est très complet, les quelques fonctions que l'on utilise sont :

- créer des géométries,
- lire des fichiers CAO (au format step) provenant d'autres logiciels de CAO,
- réaliser un maillage,
- visualiser des résultats sur un maillage.

L'installation se fait via synaptic pour Ubuntu, et en téléchargement via le site web officiel pour windows et Mac OS : <http://geuz.org/gmsh/>. Le site web contient des tutoriaux, des documentations et d'autres informations utiles.

Exercice XIV.1 Créer le fichier texte plaque-trouee.geo suivant, puis effectuer un maillage de triangles à 3 nœuds (h est la taille des éléments).

```
h = 10;
Point(1) = {0, 50, 0, h};
Point(2) = {0, 200, 0, h};
Point(3) = {50, 0, 0, h};
Point(4) = {100, 0, 0, h};
Point(5) = {100, 200, 0, h};
Point(6) = {0, 0, 0};
Line(1) = {1, 2};
Line(2) = {3, 4};
Line(3) = {4, 5};
Line(4) = {5, 2};
Circle(5) = {3, 6, 1};
Line Loop(6) = {3, 4, -1, -5, 2};
```

XIV.5 Un exemple de calcul mené avec SILEX

XIV.5.1 Géométrie et conditions aux limites

La géométrie du piston est donnée sur la figure XIV.2 ainsi que les conditions aux limites permettant de tenir compte des deux symétries. Le matériau a pour module d'Young 200000 MPa et pour coefficient de Poisson 0,3. La face supérieure est soumise à une pression de 10 MPa. La géométrie au format step est disponible sur le lien suivant :

<http://antoinelegay.free.fr/silex/silex.html>.

Les tests ont été effectués sur une machine de 16 Go avec un processeur possédant 4 cœurs cadencés chacun à 3.4 GHz mais un seul processeur a été utilisé.

XIV.5.2 Analyse des résultats

Le deux types d'éléments tétraèdres à 4 et 10 nœuds sont testés. Les différents maillages utilisés pour les tétraèdres à 4 nœuds sont présentés sur la figure XIV.3.

La figure XIV.4 montre l'évolution du champ de déplacements pour les tétraèdres à 4 nœuds. Les figures XIV.5 et XIV.6 présentent l'évolution des contraintes de Von Mises dans les éléments

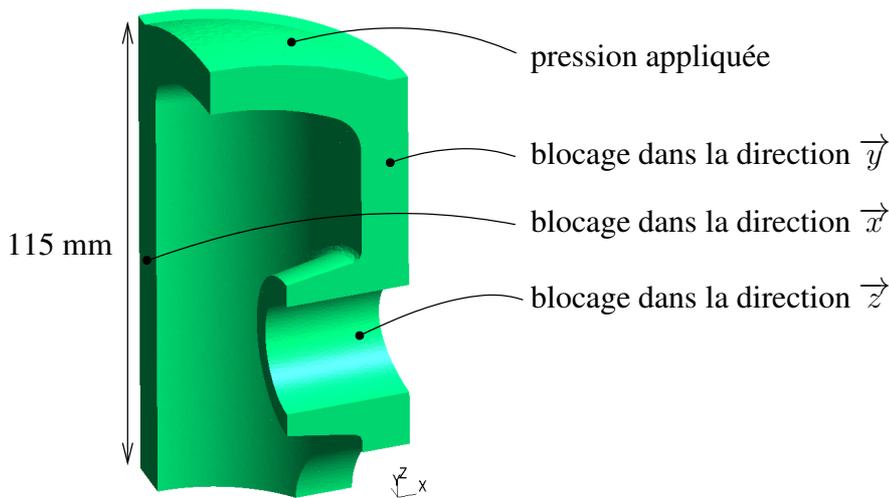


Figure XIV.2 – Géométrie et conditions aux limites pour le calcul du piston

et lissées aux nœuds du maillage pour les tétraèdres à 4 nœuds. Les contributions des éléments à l'erreur globale sont montrées sur la figure XIV.7 pour les tétraèdres à 4 nœuds.

Le déplacement converge vers une valeur maximale de 0,0245 mm (figure XIV.8(a)). Des oscillations sont observées dans l'étude de convergence de la contrainte maximale de Von Mises dans les éléments (figure XIV.8(b)) : les contraintes étant constantes dans les éléments tétraèdres à 4 nœuds, elles sont sensibles à la qualité locale du maillage. La contrainte maximale de Von Mises lissée converge vers une valeur comprise entre 80 MPa et 85 MPa (figure XIV.8(b)). L'évolution de l'erreur globale est présentée sur la figure XIV.9(a) :

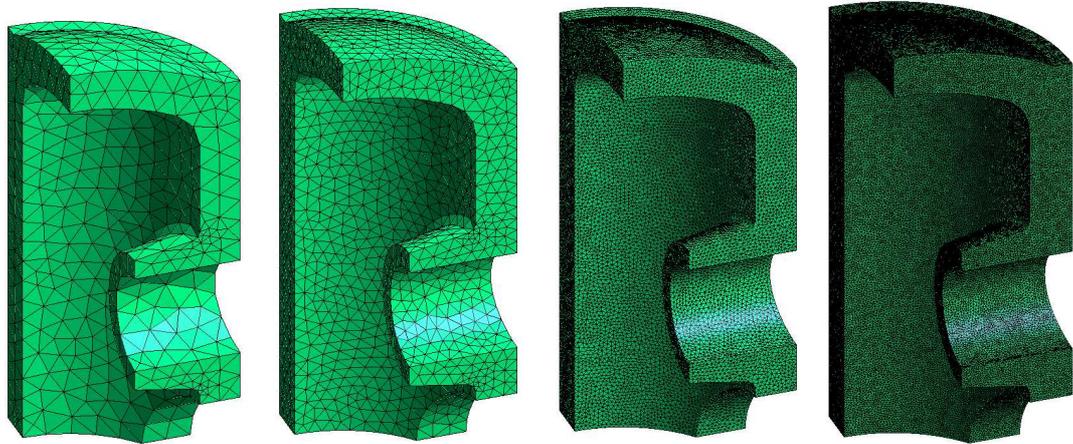
- pour les tétraèdres à 4 nœuds, la pente est de 0,56 alors que la pente théorique optimale est de 1 ;
- pour les tétraèdres à 10 nœuds, la pente est de 1,2 alors que la pente théorique optimale est de 2 ;

La figure XIV.9(b) montre l'évolution du temps de calcul en fonction du nombre de nœuds du maillage pour les deux types d'éléments. Ce temps est limité ici au temps de résolution du système pour trouver la solution en déplacement : le temps réel est plus grand en prenant en compte la construction du maillage, les étapes de lecture et d'écriture ainsi que le calcul des contraintes et de l'erreur. À nombre de nœuds égal, le temps de calcul est sensiblement le même pour les deux types d'éléments. L'évolution de ce temps est une droite en fonction du nombre de nœuds dans un diagramme log-log, il évolue donc selon une loi puissance avec le nombre d'inconnues à résoudre.

XIV.5.3 Conclusions de l'étude

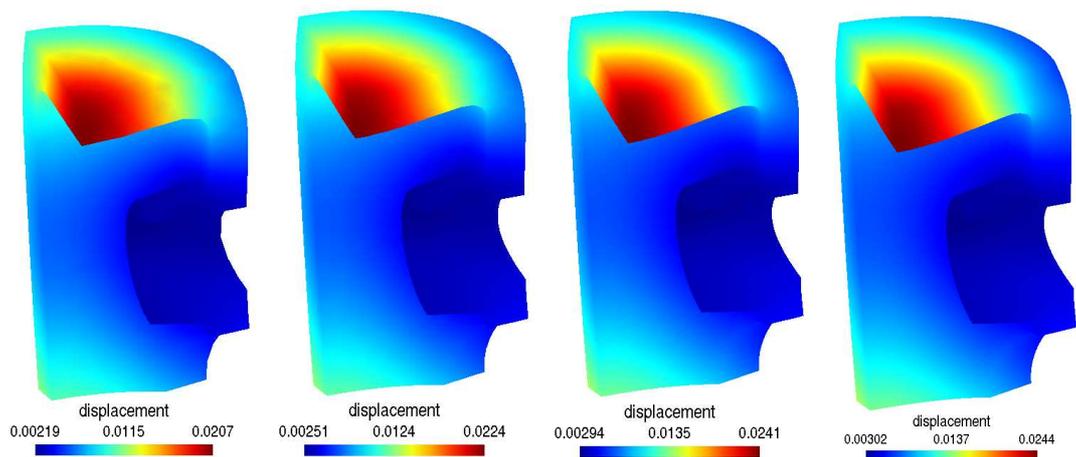
Ce problème assez sévère en terme de concentration de contrainte montre la difficulté des éléments tétraèdres à 4 nœuds à converger rapidement vers la solution. Les tétraèdres à 10 nœuds convergent plus rapidement avec des temps de calculs équivalents.

Cette étude montre les possibilités du code SILEX qui peut traiter des problèmes relativement important en terme de nombre de degrés de liberté, en fonction de l'ordinateur à disposition.



(a) 1011 nœuds, 3622 éléments (b) 3065 nœuds, 11891 éléments (c) 35021 nœuds, 154526 éléments (d) 161639 nœuds, 865338 éléments

Figure XIV.3 – Maillages de tétraèdres à 4 nœuds utilisés



(a) Maillage *a*, Déplacement maximal de 0,0207 mm (b) Maillage *b*, Déplacement maximal de 0,0224 mm (c) Maillage *c*, Déplacement maximal de 0,0241 mm (d) Maillage *d*, Déplacement maximal de 0,0244 mm

Figure XIV.4 – Évolution du champ de déplacement (tétraèdres à 4 nœuds)

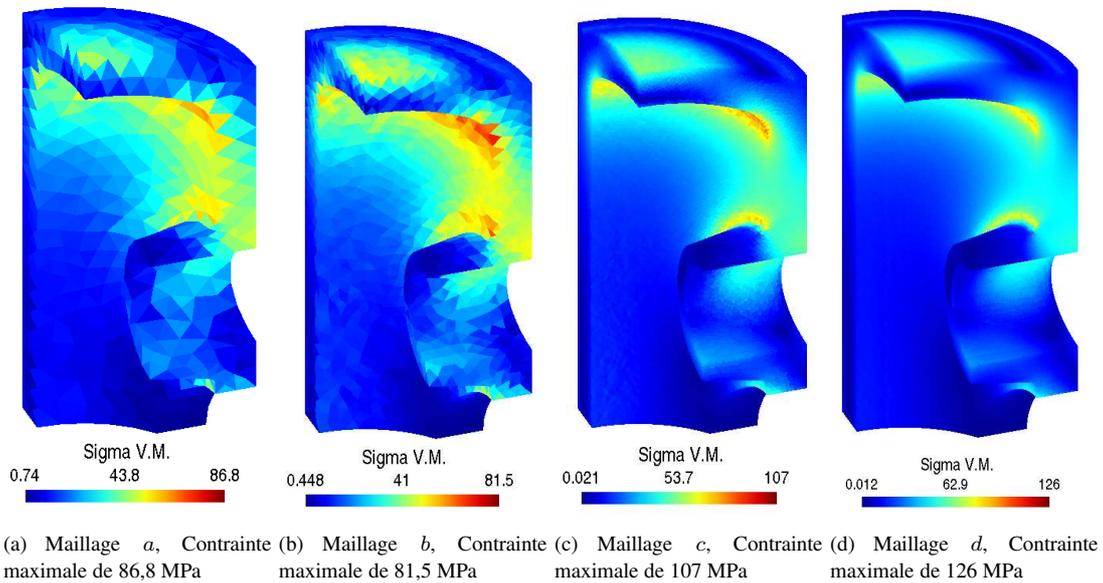


Figure XIV.5 – Évolution de la contrainte de Von Mises dans les éléments (tétraèdres à 4 nœuds)

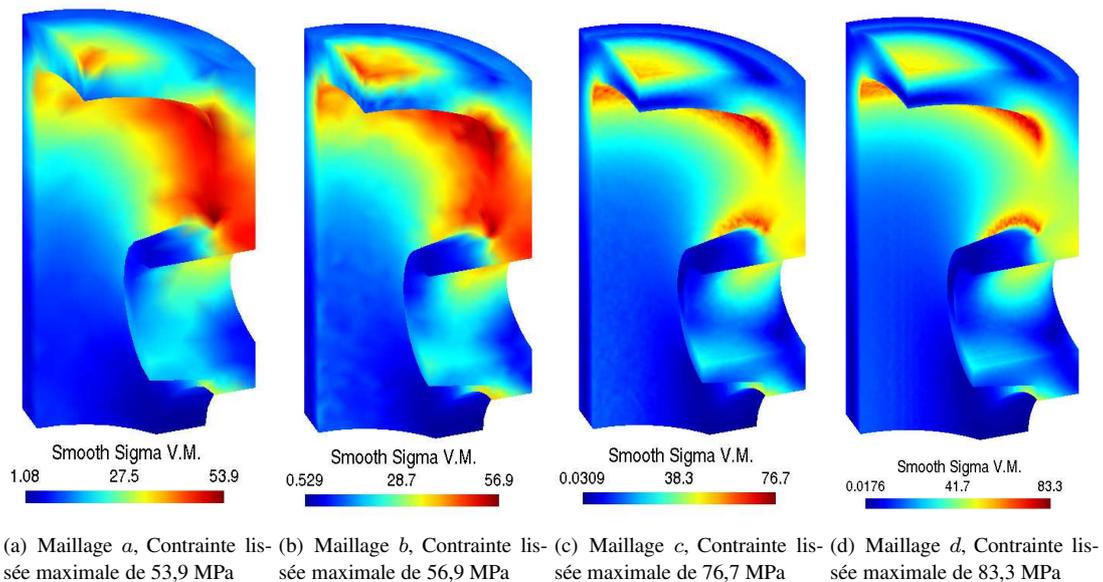
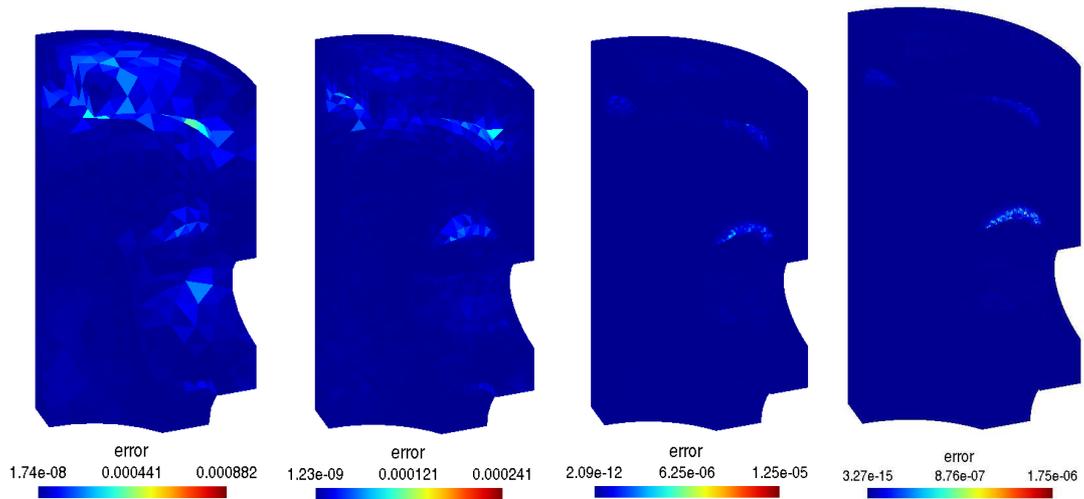
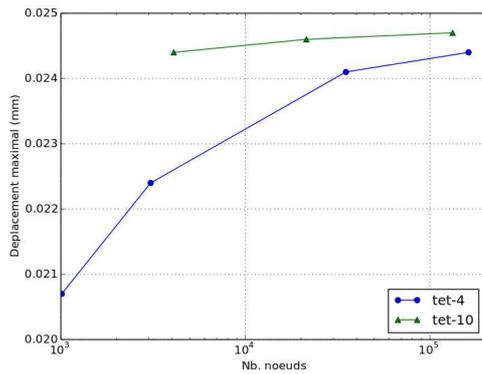


Figure XIV.6 – Évolution de la contrainte lissée de Von Mises (tétraèdres à 4 nœuds)

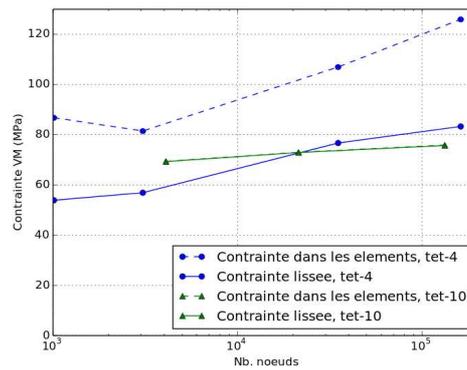


(a) Maillage *a*, erreur globale de 30,4% (b) Maillage *b*, erreur globale de 23,4% (c) Maillage *c*, erreur globale de 12,4% (d) Maillage *d*, erreur globale de 8,5%

Figure XIV.7 – Évolution de la contribution des éléments à l’erreur globale (tétraèdres à 4 nœuds)

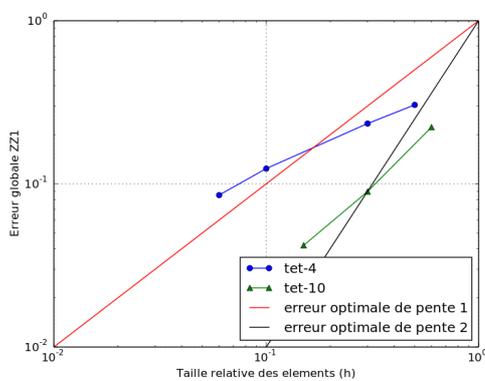


(a) Évolution du déplacement maximal

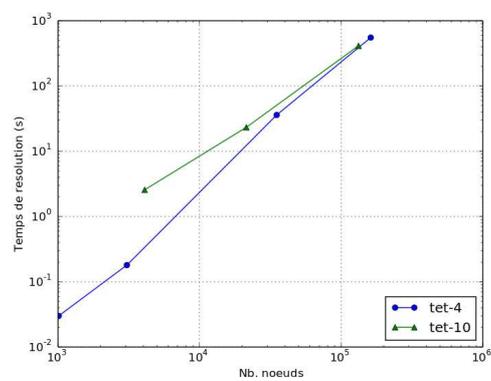


(b) Évolution de la contrainte maximale de Von Mises

Figure XIV.8 – Étude de convergence en contrainte et en déplacement



(a) Évolution de l’erreur globale



(b) Évolution du temps de résolution du système

Figure XIV.9 – Étude de convergence (tétraèdres à 4 nœuds)