REPUBLIQUE DU SENEGAL

UNIVERSITE CHEIKH ANTA DIOP DE DAKAR

Ecole Supérieure Polytechnique
Centre de THIES

DEPARTEMENT GENIE CIVIL

PROJET DE FIN D'ETUDES

EN VUE DE L'OBTENTION DU DIPLOME D'INGENIEUR DE CONCEPTION

Titre : Utilisation des éléments surfaciques dans L'analyse structurale

Auteur : **Ousmane Arëm GOUDIABY**
Baba LY

Directeur interne : **Dr. Moustapha NDIAYE**

Année académique : 2005-2006

DEDICACES

Nous dédions ce travail à nos pères, nos mères, nos frères et sœurs ainsi qu'à tous nos amis.

REMERCIEMENTS

Au terme de notre projet nous aimerions exprimer toute notre gratitude à tous ceux qui de près ou de loin nous ont apporté leur aide. Nous exprimons toute notre reconnaissance à notre directeur de projet, **le docteur Moustapha NDIAYE** qui nous a donné la possibilité d'effectuer ce travail dans le domaine des structures. Nous tenons à le remercier pour l'aide précieuse apportée durant nos discussions, pour ses conseils, ses encouragements et sa patience.

Nos remerciements s'adresse également à l'ensemble du corps professoral de l'école Supérieure Polytechnique (centre de Thiès) pour leurs enseignements, à nos camarades de promotion, ainsi que l'ensemble des étudiants de l'école.

Sommaire

Le but de ce travail est d'une part, de développer la méthode des éléments finis (MEF) et son application dans l'analyse structurale des éléments surfaciques, de l'autre d'acquérir une base de connaissances pour le développement d'une interface graphique basée sur la programmation orientée objet (POO) pourvue d'un code de calcul par éléments finis (CCEF).

La méthode des éléments finis est une procédure numérique qui permet de résoudre la plupart des problèmes que rencontre l'ingénieur, particulièrement ceux associés à l'analyse des contraintes, de transmission de chaleur, d'électromagnétisme et d'écoulement de fluide.

En général, le problème majeur que l'on rencontre dans l'application de cette méthode est de trouver un modèle mathématique adéquat pour représenter la structure physique à étudier et aussi le stockage des données informatiques vu que la méthode génère parfois des matrices de grande taille. Un modèle mathématique pourrait s'agir d'une équation aux dérivées partielles (EDP) avec des conditions aux limites et des conditions initiales.

Avec l'avènement des langages de POO, il existe actuellement des programmes de calcul structural (PCS) permettant de résoudre des systèmes linéaires de grande taille en un temps record.

Le travail que nous avons effectué constitue un point de départ pour la construction de programmes de calcul structural par éléments finis et l'essentielles de nos recommandations porte sur celle-ci.

Mots clés : MEF, éléments surfaciques, POO, CCEF, EDP, PCS

TABLE DES MATIÈRES

Page

DEDICACES.....I
REMERCIEMENTS.....II
SOMMAIRE.....III
LISTE DES FIGURES.....VII
LISTE DES SYMBOLES ET ABREVIATIONS.....IV
Avant-propos.....XI

Introduction.....1

Partie I. Fondements théoriques de la méthode des éléments finis

Chapitre : **page**

I.1 Rappel de la mécanique des milieux continus.....2
I. 2 Méthode de résolution numérique.....5
I.3. Formulation variationnelle.....6
I.4 Principe des travaux virtuels.....9
I.5 Energie potentielle totale.....10
I.6 Les fondements de la méthode-P.....15
I.7 Intégration numérique.....18
I.7.1 Méthode de Newton Cotes.....18
I.7.2 Méthode de GAUSS.....19
I.8 Modélisation des éléments surfaciques.....21
I.8.1 Les fonctions de forme ou fonctions d'interpolation.....21
I.8.2 Fonction de forme type Lagrange.....24
I.8.3 Fonction de forme type Serendip.....26
I.8.4 Eléments de membrane.....28
I.8.5 Eléments finis de membrane.....30
I.8.6 Eléments de plaque.....32
I.8.6.1 Equations cinématiques.....32

I.8.6.2 Equations d'équilibre.....34
I.8.7 Eléments de coque.....37
I.8.7.1 Hypothèses de la théorie des coques.....38
I.9 Procédure de résolution.....39
I.9.1Démarche de résolution par éléments finis.....39
I.9.2 Post-traitement.....41

Partie II : Interface graphique et code de calcul par éléments finis

I. Rappel de la programmation orientée objet POO.....41
I. 1 Notion de classe..... 42
I. 2 La notion d'héritage 43
I. 3 Notion d'encapsulation des données..... 44
II Présentation du problème.....45
II. 1Cahier de charge..... 45
II. 2 Différents modules d'un solveur éléments finis 45
III Déroulement d'un calcul statique linéaire.....46
III. 1 PROCEDURE DETAILLEE DU CALCUL 47
III. 2 Traduction des données utilisateur en données éléments finis 47
III. 3 Génération des matrices élémentaires 48
III. 4 Assemblage..... 50
III. 5 Inversion.....53
III. 6 Traitement des fixations.....53
III. 7 Détermination des déplacements.....55
III. 8 Calcul des contraintes.....56
IV. Les grands concepts de l'interface graphique d'un CCEF.....56
IV.1 Les données d'entrées d'un problème.....57
IV. 2 Vue d'ensemble de la résolution d'un problème.....57
IV. 3 La géométrie.....59
IV. 4 Entité géométrique.....60
IV. 5 Relation Géométrie / Matériaux.....61

IV. 6 Relations Entités géométriques / Conditions aux limites.....62

IV. 7 Le maillage.....63

IV. 8 Hiérarchie d'éléments.....66

IV.8.1 La notion d'élément.....64

IV. 9 Les connectivités de base.....68

V Les champs.....70

V. 1 Les champs de transformation géométrique.....70

V. 2 Les champs éléments finis avec DDL.....71

V. 3 Les lois de comportement.....75

V. 4 Relations Géométrie/Entités Géométriques/Maillage/Champ.....76

V. 5 Formulation variationnelle et résolution d'un problème.....77

V. 5. 1 Détails de la résolution d'un problème.....77

V. 5. 1. 1 Lecture des données.....78

V. 5. 1. 2 Assemblage et résolution du système global.....78

V. 5. 1. 3 Exportation des résultats.....79

V. 5. 1. 4 Post-traitement.....80

Conclusion.

BIBLIOGRAPHIE.....81

ANNEXES.....82

LISTE DES FIGURES

Page

Partie I

Fig.I.1 : Position des nœuds pour les carrés de Lagrange.....25
Fig.I.2 Mode bulle associée à un nœud interne 26
Fig.I.3 : Fonction de forme associée à un nœud sommet.....26
Fig. I.4: Position des noeuds pour les carrés de Serendip.....27
Fig. I.5: Quadrilatère non rectangulaire.....26
Fig.I.6 : feuillet moyen et axes d'une membrane.....29
Fig.I.7 : Répartition des contraintes dans une membrane.....30
Fig.I.8 : éléments du premier degré.....31
Fig.I.9 : éléments du second degré.....31
Fig.I.10 : répartition des contraintes dans une plaque.....34
Fig. I.11: efforts résultant sur une plaque.....34
Fig.I.12 : Feuillet moyen d'une coque.....37
Fig. I.13: Membrane et plaque.....38

Partie II

Fig.II.1 : élément de référence rectangulaire.....50
Fig. II.2 : Procédure d'assemblage des éléments de la structure.....53
Fig. II.3: Vue globale de la résolution d'un problème.....58
Fig. II.4 : exemple de géométrie.....60
Fig.II.5 : Hiérarchie des classes de dessin de la structure.....60
Fig.II.6 : une géométrie avec des maillages différents.....61
Fig. II.7: entité géométrique.....62
Fig.II.8 : relation entre la géométrie, les entités et les matériaux.....63
Fig.II.9 : relation entre la géométrie, les entités et les conditions aux limites.....64
Fig. II.10 : Le contenu de la classe «Maillage».....65
Fig. II.11: Routines d'informations sur un élément 66
Fig.II.12 : connectivité entre les arêtes et les nœuds.....68
Fig.II.13 : connectivité entre les faces, arêtes et sommets.....70
Fig. II.14: les champs de transformation géométrique.....72

Fig. II.15: exemples de champs d'éléments finis.....73
Fig. II.16: Champs d'éléments finis combinés à un champ de transformation géométrique...74
Fig. II.17: Champs linéaires pour tous les types d'élément.....75
Fig. II.18: champs quadratiques pour tous les types d'éléments.....75
Fig.II.19: interpolation mixte : Quad-Lin.....76
Fig. II.20: relations entre la géométrie, les entités, les maillages et les différents champs.....78

LISTE DES SYMBOLES ET ABREVIATIONS

[ ]	Matrice rectangulaire ou carré,
{ }	Vecteur Colonne,
$[\ ]^{-1}$	Inverse Matrice,
$[\ ]^{T}$	Matrice transposé
Ω	Domaine délimitant une structure
U	Champ de déplacement
$[N]$	Matrice des fonctions d'interpolation
$\{d\}$	Vecteur des degrés de libertés indépendantes de la structure
$\{d_i\}$	Vecteur déplacement au nœud i
$\{\alpha\}$	Vecteur de coordonnées généralisées
$[C]$	Matrice des coordonnées nodales
N_i	Fonction d'interpolation au nœud i
$\sigma(x)$	Champ de contraintes
$\varepsilon(x)$	Champ de déformations
σ_{zz}	Contrainte normale au plan du feuillet moyen
$\tau_{xz}, \tau_{zx}, \tau_{zy}$	Contraintes tangentielles
ε_{zz}	Déformation suivant z
E	Module d'élasticité de YOUNG
ν	Coefficient de poisson
T	Epaisseur de l'élément surfacique
$[K]$	Matrice de raideur de la structure
$[K_e]$	Matrice de raideur d'un élément
f_v	Force de volume
$[L_i]$	Matrice de localisation globale
$[R]$	Vecteur des charges appliquées sur la structure

P	Charges ponctuelles
Q	Charges réparties
I / O	Entrée /sortie
L	Opérateur de dérivation
$[B]$	Matrice déformation déplacements
K_{ij}	Éléments de la matrice de raideur de la structure
d_f	Déplacements fixés
d_l	Déplacements libres
DLL	Degrés de liberté libres
EDP	Equations aux dérivées partielles
MEF	Méthode des éléments finis
POO	Programmation orientée objet
CCEF	Codes de calcul par éléments finis
PCS	Programme de calcul structural
CL	Conditions aux limites
2D	Deux dimensions

Avant-propos

Pourquoi ce projet ?

L'Ecole Supérieure Polytechnique (E.S.P.) est un établissement qui regroupe, depuis la réforme de 1994, l'ex-E.N.S.U.T., l'ex-E.P.T. et l'ex-E.N.S.E.P.T.

Elle est rattachée à l'Université Cheikh Anta DIOP de Dakar et comporte deux centres : le centre de Dakar et le centre de Thiès.

L'E.S.P. est constituée de cinq (5) départements répartis dans les deux centres comme suit :

- ✓ Centre de Dakar :
  - ➢ Département Génie Chimique ;
  - ➢ Département Génie Civil (formation continue) ;
  - ➢ Département Génie Electrique ;
  - ➢ Département de Gestion ;
  - ➢ Département Génie Informatique ;
  - ➢ Département Génie Mécanique (D.U.T.) ;
- ✓ Centre de Thiès :
  - ➢ Département Génie Civil (D.U.T. et D.I.C.) ;
  - ➢ Département Génie Mécanique (D.I.C.)

L'E.S.P. a pour vocation la formation de techniciens supérieurs (D.U.T.), d'ingénieurs technologues (D.I.T.) et d'ingénieurs de conception (D.I.C.) mais aussi la recherche à travers le troisième cycle. Les durées de formation sont de deux ans pour le D.U.T., de trois ans et demi en formation continue pour le D.I.T. et de trois ans pour le D.I.C.

A la fin du cycle d'ingénieur de conception, l'élève ingénieur est appelé à mener un projet de fin d'études, dont celui-ci, sous la direction de ses professeurs et éventuellement de personnes externes. Ce projet lui permettrait de mettre en application les différentes connaissances théoriques et pratiques acquises lors de son cycle.

INTRODUCTION

Les éléments surfaciques trouvent leur application dans plusieurs domaines, plus particulièrement dans le domaine du génie civil. Ils sont souvent utilisés comme parement, dalle, radier et plancher dans le bâtiment, comme dômes dans les grands ouvrages d'art, et aussi comme tablier de pont.

L'impossibilité de trouver analytiquement la solution aux problèmes mécaniques de telles structures, nous pousse alors à utiliser des méthodes numériques pour approcher la solution.

La méthode des éléments finis est utilisée pour la résolution numérique d'équation aux dérivées partielles. C'est une méthode qui offre l'avantage par rapport à la méthode des différences finies, de pouvoir traiter sans trop de difficultés supplémentaires toute géométrie ainsi que d'augmenter la précision des résultats au prix d'efforts de programmation raisonnables. C'est ainsi que des logiciels de calculs scientifiques empruntant de plus en plus la voie des langages naturels ont connu un essor remarquable. Des outils numériques de plus en plus conviviaux et performants permettent de résoudre des problèmes réels de plus en plus complexes dans divers domaine comme le génie civil, le génie mécanique, les mathématiques appliquées, la géophysique etc.

Les objectifs de nos travaux concernent le développement de la méthode des éléments finis appliquée aux éléments surfaciques, l'acquisition d'une base de connaissances pour le développement d'un code de calcul par éléments finis (CCEF), contenant un moteur de calcul reproduisant la MEF et fondé sur une nouvelle architecture .

Dans ce projet, nous proposons de développer la méthode des éléments finis aux problèmes d'analyse structurale des éléments surfaciques dans un environnement orienté objet.

La première partie constitue la description générale de la MEF appliquée aux éléments surfaciques (membranes, plaques, coques) et dans la deuxième, nous traiterons du pré développement d'une interface graphique pourvue d'un code de calcul par éléments finis.

Partie I. Fondements théoriques de la méthode des éléments finis

De très nombreux problèmes physiques s'expriment sous forme d'équations aux dérivées partielles soumises à des conditions aux limites particulières et la résolution de ces équations est souvent impossible mathématiquement d'où le recours à la méthode des éléments finis pour une discrétisation du problème. C'est le cas pour les problèmes d'élasticité linéaire en particulier pour les éléments surfaciques dont les caractéristiques seront étudiées au cours de cet exposé. Nous procéderons tout d'abord à un rappel de mécanique des milieux continus puis nous mettrons en évidence l'ensemble de la théorie sous-jacente à la méthode des éléments finis et expliciterons les différentes étapes de la résolution d'un problème d'élasticité linéaire par la méthode des éléments finis.

I.1 Rappel de la mécanique des milieux continus

Le concept de milieu continu est une modélisation physique macroscopique issue de l'expérience courante, dans la formulation mathématique classique de ce concept, un système mécanique est représenté par un volume Ω constitué, au niveau différentiel, de particules $d\Omega$. L'état géométrique de ces particules est caractérisé par la seule connaissance de leur position. Au cours de l'évolution du système les particules initialement voisines demeurent voisines.
L'analyse de l'évolution du comportement du système se fait en déterminant la position dans l'espace des particules du système. La représentation mathématique de ce comportement, en utilisant les coordonnées cartésiennes, aboutit aux Equations aux Dérivées Partielles (EDP).

Résoudre un problème d'élasticité c'est :

- Déterminer le vecteur déplacement en tout point de la structure, ce qui pour un solide en 3-D représente les 3 inconnues du champ de déplacement noté u(x) :

$$u = \begin{Bmatrix} u(x,y,z) \\ v(x,y,z) \\ w(x,y,z) \end{Bmatrix} \qquad (1.1)$$

- Déterminer le tenseur de déformations en tout point de la structure, ce qui donne les 6 inconnues du champ des déformations noté ε(x) :

$$[\varepsilon(x)] = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33} \end{bmatrix} \qquad (1.2)$$

- Déterminer le tenseur de contraintes en tout point de la structure, ce qu représente les 6 inconnues du champ des contraintes noté σ(x) :

$$[\sigma(x)] = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \qquad (1.3)$$

Ce qui nous fait un total de 15 inconnues à déterminer. Cette analyse se fait grâce aux équations de base de la théorie de l'élasticité linéaire dans le cadre des petites déformations. Ces équations proviennent de différentes relations que sont celles qui lient les déformations aux déplacements ainsi que les relations contraintes–déformations ; nous avons :

- Les relations de compatibilité qui lient les déplacements (3 composantes) et les déformations (6 composantes) :

$$\varepsilon_x = \frac{\partial u}{\partial x} \; ; \; \varepsilon_y = \frac{\partial v}{\partial y} \; ; \varepsilon_z = \frac{\partial w}{\partial z} \qquad (1.4)$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \quad \gamma_{zy} = \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \quad \gamma_{xz} = \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \qquad (1.5)$$

- La loi de comportement ou loi de Hooke qui lie le tenseur de contraintes au tenseur des déformations, elle s'écrit sous la forme :

$$\sigma = H\left(\varepsilon - \varepsilon_0\right) + \sigma_0 \qquad (1.6)$$

Où σ et ε sont respectivement le tenseur de contraintes et de déformations et **H** est la matrice qui définit le comportement du solide et dépend du milieu, pour un milieu homogène isotrope, elle s'écrit sous la forme :

$$H = \begin{bmatrix} H_\sigma & 0 \\ 0 & H_T \end{bmatrix} \qquad (1.7)$$

Avec

$$\mathbf{H}_\sigma = \frac{E}{(1-\nu^2)}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \quad \text{et} \quad H_T = G\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Avec $G = \frac{E}{2(1+\nu)}$; E : module d'élasticité et ν : coefficient de poisson

Nous avons donc la matrice 6x6 du type :

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ 2\varepsilon_{yz} \\ 2\varepsilon_{xz} \\ 2\varepsilon_{xy} \end{bmatrix} \qquad (1.8)$$

- Les équations d'équilibre pour un solide quelconque s'écrivent sous la forme :

$$\begin{aligned}
\frac{\partial \sigma_{xx}}{\partial x}+\frac{\partial \sigma_{xy}}{\partial y}+\frac{\partial \sigma_{xz}}{\partial z}+f_x=0\\
\frac{\partial \sigma_{xy}}{\partial x}+\frac{\partial \sigma_{yy}}{\partial y}+\frac{\partial \sigma_{yz}}{\partial z}+f_y=0\\
\frac{\partial \sigma_{xz}}{\partial x}+\frac{\partial \sigma_{zy}}{\partial y}+\frac{\partial \sigma_{zz}}{\partial z}+f_z=0
\end{aligned} \qquad (1.9)$$

ou encore sous forme matricielle :

$$\{div[\sigma]^T\}+\{f_v\}=0 \qquad (1.10)$$

Avec σ_{ij} étant les composantes du tenseurs de contraintes et f_x, f_y, f_z les composantes du vecteur chargement dans un repère cartésien (x, y, z).

I. 2 Méthode de résolution numérique

Bien que la connaissance de ces équations aux dérivées partielles soit parfois ancienne, seul très peu de cas peuvent être résolus par les mathématiques classiques. On remplace donc le problème continu par un problème approché en discrétisant la structure et obtenir des solutions approchées via des méthodes numériques. La méthode numérique la plus utilisée de nos jours reste celle des éléments finis.

La méthode englobe trois domaines principaux :

- Les méthodes de discrétisation qui permettent de transformer un problème continu en une approximation discrète,
- les méthodes variationnelles qui permettent de transformer une équation aux dérivées partielles (EDP) en une forme approchée « variationnelle ».
- les méthodes numériques qui permettent de résoudre les systèmes d'équations linéaires, non linéaires et de rechercher des valeurs propres .....

Le tout allié à des moyens de calcul qui exécutent les instructions de plus en plus rapidement actuellement.

I.3. Formulation variationnelle

Dans le cadre de la mécanique des milieux continus, l'équation représente l'équilibre des forces et des moments qui agissent sur un élément de matière. Elle s'écrit sous la forme d'une équation aux dérivées partielles dont l'inconnue est le champ de déplacement.
Mathématiquement, l'équation est difficile, voire même impossible à résoudre dans le cas général, dans ce contexte, la formulation variationnelle de l'EDP, apparaît comme une forme équivalente de l'équation, mais qui a pour avantage de pouvoir par la suite être résolue de manière approchée, ce qui sera le cas dans la méthode des éléments finis.
Dire qu'un solide sous l'action d'un ensemble de forces est en équilibre, équivaut à dire que la puissance totale mise en jeu par ces forces lors d'un déplacement quelconque, est nulle, en d'autres termes :

$$\left\{\sum \overrightarrow{F_i} = \vec{0}\right\} \Leftrightarrow \left\{\overset{\bullet}{P} = \sum \overset{\bullet}{P_i} = \left(\sum \overrightarrow{F_i}\right).\overrightarrow{\delta U} = 0\right\} \quad \forall \delta U \qquad (1.11)$$

Avec δU étant un déplacement virtuel.

Cette équation faisant intervenir les puissances est appelée forme variationnelle des équations d'équilibre. Elle est identique à l'équation originale d'équilibre, tant qu'elle est vérifiée pour tout déplacement virtuel δU .
Mais dans le cadre de l'équilibre interne des forces de cohésion de la matière, représentées par les contraintes, la puissance virtuelle mise en jeu lors d'un déplacement virtuel est une fonction d'une part du déplacement virtuel, et d'autre part du déplacement réel. En effet, les contraintes peuvent être calculées à partir des déformations via la loi de comportement du matériau, et les déformations peuvent être calculées à partir du champ de déplacement.
Ainsi la seule inconnue de notre problème variationnelle demeure le champ de déplacements $\vec{U}$. En définitif on peut écrire :

$$\overset{\bullet}{P} = P\left(\overset{\bullet}{\vec{U}}, \overrightarrow{\delta U} = 0\right) \quad \forall \overrightarrow{\delta U} \qquad (1.12)$$

Supposons maintenant que l'on cherche une forme approchée de $\vec{U}$ à l'aide d'une discrétisation par éléments finis $\vec{U}_a$. On a vu que le champ approché $\vec{U}_a$ se calcule à partir de

la valeur du déplacement en chaque nœud $\vec{U}_r$, l'indice r étant le numéro du nœud. Dans ce cas, la puissance virtuelle est une fonctionne des valeurs aux nœuds :

$$\overset{*}{P} = P\left(\overrightarrow{\overset{*}{U}_r}, \overrightarrow{\delta U} = 0\right) \quad \forall \overrightarrow{\delta U} \qquad (1.13)$$

Lorsque l'on connaît toutes les valeurs aux nœuds, appelées valeurs nodales, le problème est résolu. Si N est le nombre de nœuds, le problème comporte 3xN inconnues (pour un modèle 3D), que l'on appelle degrés de liberté du problème.

Sachant que chaque déplacement virtuel δU conduit à une équation qui doit être vérifie pour satisfaire l'équilibre, il nous reste à déterminer 3xN déplacements virtuels possibles.

Soit un nœud donne, par exemple le numéro s, et choisissons un déplacement δU_s^a tel que ce déplacement soit nul au niveau de tous les nœuds sauf au nœud s où il vaut la valeur 1. Pour la composante a (=1 ou 2 ou 3), et qu'il passe d'une manière continue de 1 à 0 du nœud 5s aux autres nœuds. Il est clair que cette méthode de construction de déplacement virtuel permet de déterminer 3xN déplacements virtuels, élémentaires. A l'aide de ces déplacements, nous allons pouvoir générer 3xN équations, dont les inconnues sont les déplacements aux nœuds.

Dans le cas où ces équations sont linéaires en inconnues (c'est-à-dire forment une combinaison linéaire des inconnues) on parle de problème linéaire, c'est le cas par exemple de l'étude de la déformation élastique d'un corps (ou structure) en petite déformation, sinon on parle de problème linéaire.

Les problèmes linéaires sont évidemment les plus simples à résoudre. Le système d'équation s'écrit alors sous une forme matricielle équivalente à :

$$[K]\{U\} = \{F\} \qquad (1.14)$$

Où :

- (U) est le vecteur déplacement inconnu, ses composantes sont en fait les composantes des déplacements inconnus pour chaque nœud du maillage,
- (F) est le vecteur second membre connu, ses composantes sont en fait les composantes des forces imposées en chaque nœud.

$$F_i = \langle f, P_i \rangle - k(u_d, P_i) \qquad (1.15)$$

[K] est la matrice de raideur, c'est une matrice de composantes connues, fonction de le forme de la pièce calculée et de la loi de comportement : élastique, plastique…

$$K_{ij} = k(P_i, P_j) \qquad (1.16)$$

ou dans un cas plus général

$$F_i = \langle f, P_i \rangle + \int_{\Omega} tr(\sigma^{th}.\varepsilon(P_i))dw - k(u_d, P_i) \qquad (1.17)$$

Résoudre le problème équivaut alors à résoudre ce système matriciel par exemple en inversant la matrice à condition que le déterminant de [K] soit non nul :

$$\{U\} = [K]^{-1} \{F\} \qquad (1.18)$$

Dans le cas d'un problème non linéaire, par exemple lors de l'étude de grandes transformations, en général on utilise une méthode de résolution numérique, par exemple la méthode de NEWTON-RAPHSON qui consiste à remplacer le problème original par une succession de problèmes linéaires. Il est clair que la résolution est alors plus complexe et que les temps de calcul plus longs. Néanmoins, la prise en compte de ce type de problème tend aujourd'hui à se généraliser, due en particulier aux grandes vitesses des calculateurs modernes.

Cette formulation variationnelle découlant du Principe des Puissances Virtuelles est connue sous le nom de méthode de GALERKIN.

On se donne n fonctions de base $\{P_i(x)\}$ appartenant au cinématiquement admissible à zéro et on cherche la solution du problème comme une combinaison linéaire de ces fonctions, dans le cas où il existe des déplacements imposés on rajoute une fonction les réalisant c'est-à-dire :

$$\overset{*}{u}(x) = P_i(x) \qquad (1.19)$$

$\in$ C.A. (champ de déplacements cinématiquement admissibles)

$$\overset{*}{u}(x) = P_i(x) \qquad (1.20)$$

En supposant que Le problème d'élasticité étant équivalent à la formulation variationnelle nous obtenons alors :

$$k\,(u\,(x)\,,\,P_i\,(x)) \;=\; P_{donné}\,(P_i\,(x)) \qquad (1.21)$$

Il suffit alors de faire varier i de 1 à n. Nous obtenons alors un système linéaire de n équations à n inconnues. Ce système peut s'écrire sous forme matricielle :

$$[K][U] = [F] \qquad (1.22)$$

La matrice K est symétrique. L'équation mise sous sa forme matricielle correspond à la forme générale d'un problème discrétisé. En effet, nous sommes passé d'un problème continu à un problème discrétisé, de l'étude u(x, y, z) à l'étude de n inconnues U_i.

En résume, il faut retenir que la formulation variationnelle alliée aux techniques de discrétisation permet de transformer l'équation originale d'équilibre, en un système d'équations, linéaires dans les cas simples, qui après résolution fournit les déplacements aux nœuds du maillage.

On remarque que plus le nombre de nœuds est important, plus la taille du système linéaire à résoudre est importante. Couramment la dimension des systèmes à résoudre atteint plusieurs milliers voire plusieurs dizaines de milliers.

I.4 Principe des travaux virtuels

Une autre forme de formulation variationnelle est obtenue en utilisant la méthode des résidus pondérés :

$$W = -\int_v \langle u^* \rangle \left(\{ div[\sigma] \} + \{ f_v \} \right) dV = 0 \qquad (1.23)$$

Pour tout champ de déplacements virtuels.
En intégrant par parties nous obtenons la forme faible de W :

$$W = W_{int} - W_{ext} = 0 \quad \forall u^*(x, y, z) \qquad (1.24)$$

Avec

$$W_{int} = \int_V \left(\frac{\partial u^*}{\partial x}.\sigma_{xx} + \frac{\partial v^*}{\partial y}.\sigma_{yy} + \frac{\partial z^*}{\partial z}.\sigma_{zz} + \left(\frac{\partial u^*}{\partial y} + \frac{\partial v^*}{\partial x} \right).\sigma_{xy} + \left(\frac{\partial u^*}{\partial z} + \frac{\partial w^*}{\partial x} \right).\sigma_{xz} + \left(\frac{\partial v^*}{\partial z} + \frac{\partial zw}{\partial y} \right).\sigma_{zy} \right) dV$$

$$W_{int} = \int_V tr\left(\left[D^* \right] \left[\sigma \right] \right) dV = \int_V \left\langle D^* \right\rangle \left\{ \sigma \right\} dV \qquad (1.25)$$

Et

$$W_{ext} = \int_V \left(\left\langle u^* \right\rangle . \left\{ f_V \right\} \right) dV + \int_{S_f} \left(\left\langle u^* \right\rangle . \left\{ f_S \right\} \right) dS + \int_{S_u} \left(\left\langle u^* \right\rangle . \left\{ \sigma(n) \right\} \right) dS \quad 1.26$$

Où :

$\left\langle u^* \right\rangle = \left\langle u^* v^* w^* \right\rangle$ est le champ déplacement virtuel

$\left\langle f_v \right\rangle = \left\langle f_x f_y f_z \right\rangle$ représente les forces volumiques qui s'appliquent au niveau du volume du solide V

$\left\langle \sigma(n) \right\rangle = \left\langle \sigma_{xn} \sigma_{yn} \sigma_{zn} \right\rangle$ représente les contraintes

$\left\langle f_s \right\rangle = \left\langle f_{sx} f_{sy} f_{sz} \right\rangle$ sont les forces surfaciques sur la partie S_f de la frontière du solide

$$\left\langle D^* \right\rangle = \left\langle \frac{\partial u^*}{\partial x} \quad \frac{\partial v^*}{\partial y} \quad \frac{\partial w^*}{\partial z} \quad \frac{\partial u^*}{\partial y} + \frac{\partial v^*}{\partial x} \quad \frac{\partial u^*}{\partial z} + \frac{\partial w^*}{\partial x} \quad \frac{\partial v^*}{\partial z} + \frac{\partial w^*}{\partial y} \right\rangle$$

Et

$$\left\langle \sigma \right\rangle = \left\langle \sigma_x \ \sigma_y \ \sigma_z \ \sigma_{xy} \ \sigma_{xy} \ \sigma_{yz} \right\rangle$$

I.5 Energie potentielle totale

Le principe des travaux virtuels traduit, sous forme intégrale, l'équilibre du solide soumis à l'action de forces de volume f_v et de surface f_s ;

$$W = \int_V \left(\left\langle \varepsilon^* \right\rangle . \left\{ \sigma \right\} \right) dV - \int_V \left(\left\langle u^* \right\rangle . \left\{ f_V \right\} \right) dV + \int_{S_f} \left(\left\langle u^* \right\rangle . \left\{ f_S \right\} \right) dS \quad (1.27)$$

$$\forall \left\{ u^* \right\} \text{ avec } \left\{ u^* \right\} = 0 \quad \text{sur } S_u$$

En introduisant dans l'expression du Principe des Travaux Virtuels la loi de Hooke généralisée, les relations déformations-déplacements et les conditions aux limites on obtient la forme intégrale suivante, fonction de u* et de u :

$$W(u^*,u)=\int_v\langle\varepsilon^*\rangle\left([H]\{\varepsilon\}+\{\sigma_0\}\right)dV-\int_v\langle u^*\rangle\{f_v\}dV-\int_{S_f}\langle u^*\rangle\{f_S\}dS=0 \quad (1.28)$$

$$\forall\{u^*\} \text{ avec } \{u^*\}=0 \text{ et } \{u\}=\{\bar{u}\} \text{ sur } S_u$$

Cette forme variationnelle sert de base pour construire les modèles déplacements éléments finis.

En admettant que les fonctions de pondération u* et les fonctions solutions u appartiennent au même espace de fonctions (même base de représentation) on obtient la formulation de type Galerkin conduisant aux modèles déplacements en éléments finis. Les déplacements virtuels u* sont alors définis comme les variations des déplacements réels :

$$u^*=\delta(u)=\delta u \qquad (1.30)$$

Où δu représente la variation de la fonction u, l'opérateur δ vérifiant les propriétés suivantes :

$$\delta(\delta u)=\delta^2 u=0 \;;\; \delta\left(\frac{\partial u}{\partial x}\right)=\frac{\partial}{\partial x}(\delta u) \quad (1.31)$$

$$\int_v \delta u dx=\delta\int_v u dx$$

L'expression de W de type Galerkin est alors :

$$W(u)=\int_v\langle\delta\varepsilon\rangle\left([H]\{\varepsilon\}+\{\sigma_0\}\right)dV-\int_v\langle\delta u\rangle\{f_v\}dV-\int_{S_f}\langle\delta u\rangle\{f_S\}dS=0 \qquad (1.32)$$

$$\forall\{\delta u\} \quad \text{avec} \quad \{\delta u\}=0 \quad \text{et} \quad \{u\}=\{\bar{u}\} \quad \text{sur} \quad S_u$$

On définit la fonctionnelle Π appelée énergie potentielle totale telle que :

$$W=\delta(\Pi(u))=\frac{\partial\Pi}{\partial u}\delta u=0 \qquad (1.33)$$

$$\text{Avec} \quad \Pi(u)=\Pi_{int}(u)-\Pi_{ext}(u) \qquad (1.34)$$

Où Π_{int} est l'énergie interne de déformation :

$$\Pi_{int}(u)=\int_v\left(\frac{1}{2}\langle\varepsilon(u)\rangle[H]\{\varepsilon(u)\}+\langle\varepsilon(u)\rangle\{\sigma_0\}\right)dV \qquad (1.35)$$

Et $-\Pi_{ext}$ est le potentiel des forces de volume et de surface (forme linéaire de u) :

$$\Pi_{ext}(u)=\int_v\langle u\rangle\{f_v\}dV+\int_v\langle u\rangle\{f_s\}dS \quad (1.36)$$

La seconde variation de Π est donnée par :

$$\delta^2\Pi(u) = \int_v \langle \delta\varepsilon \rangle [H] \{\delta\varepsilon\} dV > 0 \quad \forall \{\delta u\} \neq \{0\} \qquad (1.37)$$

Puisque [H] est définie positive et que $\{\delta^2\varepsilon\}$, δf_v, $\delta\{\sigma_0\}$, $\delta\{\sigma_{th}\}$ sont nuls par définition.

La solution {u} du problème correspond ainsi au minimum de l'énergie potentielle totale. Le principe du minimum de l'énergie potentielle s'énonce ainsi :

« Parmi tous les champs de déplacements cinématiquement admissibles celui qui rend Π minimum correspond à la solution du problème ».

En supposant que le champ de déplacements puisse se mettre sous la forme :

$$u = Nq \qquad (1.38)$$

Où q est le vecteur des degrés de liberté choisis pour représenter le comportement (coordonnées généralisées particulières), et N la matrice des fonctions d'interpolation.

La discrétisation de la forme variationnelle de l'EDP nous donne :

$$W = \sum W^e = 0 \quad \forall u^* \qquad (1.39)$$

Avec

$$W^e = \int_v \langle \varepsilon^* \rangle [H] \{\varepsilon\} dV - \int_v \langle u^* \rangle \{f_v\} dV + \int_v \langle \varepsilon^* \rangle \{\sigma_0\} dV - \int_{S_f} \langle u^* \rangle \{f_s\} dS \quad 1.40$$

$$W^e = W^e_{int} + W^e_{ext} \qquad (1.41)$$

Et

$$W^e_{int} = \int_v \langle \varepsilon^* \rangle [H] \{\varepsilon\} dV + \int_v \langle \varepsilon^* \rangle \{\sigma_0\} dV \qquad (1.42)$$

$$W^e_{ext} = \int_v \langle u^* \rangle \{f_v\} dV + \int_{S_f} \langle u^* \rangle \{f_s\} dS \qquad (1.43)$$

De façon générale l'approximation par éléments finis se traduit par :

$$\{u(\xi,t)\} = [N(\xi)]\{u_n(t)\} \qquad (1.44)$$

$$\text{Avec} \quad \{\frac{\partial}{\partial x}\} = [j]\{\frac{\partial}{\partial \xi}\} \qquad (1.45)$$

Les relations déformations déplacements s'écrivent :

$$\{\varepsilon\} = [B]\{u_n\} \qquad (1.46)$$

L'écriture de W^e sous forme matricielle nous donne :

$$W_{int}^{e} = \langle u_n^* \rangle ([k]\{u_n\} - \{f_\sigma\}) \quad (1.47)$$

$$W_{ext}^{e} = \langle u_n^e \rangle \{f_n\} \quad (1.48)$$

$$avec \quad [k] = \int_{v^e} [B]^T [H][B] dV \quad (1.49)$$

$$\{f_\sigma\} = -\int_{v^e} [B]^T \{\sigma_0\} dV \quad (1.50)$$

$$\{f_n\} = \int_{v^e} [N]^T \{f_v\} dV + \{f_\sigma\} + \int_{S_f^e} [N]^T \{f_S\} dS \quad (1.51)$$

$[k]$ Matrice de rigidité élémentaire

$\{f_n\}$ Vecteur des forces (équivalentes aux nœuds) élémentaires dues aux forces de volume et de surface

La matrice [B] est fonction de ξ, η, ζ et des termes de [j] matrice jacobienne de la transformation des coordonnées paramétriques en coordonnes cartésiennes. Les relations déformations-déplacements en coordonnées cartésiennes peuvent s'écrire sous la forme :

$$\{\varepsilon\} = [A]\{\varepsilon_\xi\} \quad (1.52)$$

Où [A] contient les termes de [J].

Pour un élément isoparametrique :

$$\begin{aligned} x &= \langle N \rangle \{x_n\} \ ; \ y = \langle N \rangle \{y_n\} \ ; \ z = \langle N \rangle \{z_n\} \\ u &= \langle N \rangle \{u_n\} \ ; \ v = \langle N \rangle \{v_n\} \ ; \ x = \langle N \rangle \{w_n\} \end{aligned} \quad (1.53)$$

Avec $\langle N \rangle = \langle N_1(\xi, \eta, \zeta) \ \ N_1(\xi, \eta, \zeta) \ .... \rangle$

$$[J] = \begin{bmatrix} \left\langle \frac{\partial N}{\partial \xi} \right\rangle \\ \left\langle \frac{\partial N}{\partial \eta} \right\rangle \\ \left\langle \frac{\partial N}{\partial \zeta} \right\rangle \end{bmatrix} [\{x_n\} \quad \{y_n\} \quad \{z_n\}] \quad (1.54)$$

$$[j] = \begin{bmatrix} j_{11} & j_{12} & j_{13} \\ j_{21} & j_{22} & j_{23} \\ j_{31} & j_{32} & j_{33} \end{bmatrix} = [J]^{-1} \quad (1.55)$$

L'expression de $\{\varepsilon_\xi\}$ est donc :

$$\{\varepsilon_\xi\} = [B_\xi]\{u_n\} \quad (1.56)$$

Avec $\langle u_n \rangle = \langle u_1 \ \ v_1 \ \ w_1 \vdots \ u_2 \ \ v_2 \ \ w_2 \ \vdots \ \ldots \rangle$ (1.57)

$$[B_\xi]=\begin{bmatrix} & \frac{\partial N_i}{\partial \xi} & 0 & 0 & & \\ \dots & 0 & \frac{\partial N_i}{\partial \xi} & 0 & \dots & i=1,2\dots \\ & 0 & 0 & \frac{\partial N_i}{\partial \xi} & & \end{bmatrix} \quad (1.58)$$

Avec $$\left\langle \frac{\partial N_i}{\partial \xi} \right\rangle = \left\langle \frac{\partial N_i}{\partial \xi} \quad \frac{\partial N_i}{\partial \eta} \quad \frac{\partial N_i}{\partial \zeta} \right\rangle$$

Ce qui nous donne :

$$\{\varepsilon\}=[B]\{u_n\} \quad ; \quad [B]=[A][B_\xi] \quad (1.59)$$

La matrice [N] est :

$$[N]=\begin{bmatrix} & N_i & 0 & 0 & & \\ \dots & 0 & N_i & 0 & \dots & i=1,2\dots \\ & 0 & 0 & N_i & & \end{bmatrix} \quad (1.60)$$

La discrétisation du problème permet d'obtenir une solution approchée dont la précision dépend de nombreux paramètres ; mais des méthodes permettant de minimiser l'erreur commise et d'accroître ainsi la précision. Parmi celle-ci deux méthodes se sont avérées très efficaces : il s'agit de la méthode-P et de la méthode-H.

La méthode-P désigne une stratégie de contrôle de l'erreur qui consiste à faire varier le degré d'interpolation des éléments tout en conservant leur taille. Elle s'oppose à la méthode-H qui consiste à faire varier la taille des éléments tout en conservant leur degré d'interpolation. Cette dernière a déjà fait l'objet d'un projet de fin d'études l'année dernière (PFE de 2004-2005).

I.6 Les fondements de la méthode-P

Les termes de la matrice élémentaire [k] et du vecteur forces sont calculés par intégration sur la géométrie d'un élément. Dans certains cas les expressions à intégrer se présentent sous forme polynomiale, le champ de déplacement s'écrit alors sous la forme :

$$u_h = \sum_{i=1}^{n} N_i a_i \qquad (1.61)$$

Une telle approximation est dite hiérarchique si le passage de n à n+1 n'altère pas les fonctions de forme N_i (i =1àn). L'avantage de ce type d'interpolation est que la discrétisation initiale peux être mise à profit pour le calcul des solutions raffinées; et que le système d'équations est mieux conditionné. Elle nous donne les solutions qui sont moins sensibles aux imprécisions numériques. Cette manière d'améliorer la précision en augmentant le degré d'approximation des éléments tout en conservant la topologie du maillage et le degré des éléments est appelée méthode-p uniforme. Elle consiste à adopter une formulation hiérarchique pour la représentation des déplacements, la matrice de raideur relative à un degré donné imbrique celles de degrés inférieurs. La procédure définissant le système éléments finis requiert la construction de la matrice de raideur $\boldsymbol{K}$ et du vecteur des forces généralisées $\{f\}$ Comme $\boldsymbol{K}$ et $\{f\}$ sont de forme intégrale, ils peuvent être obtenus par assemblage des matrices et des vecteurs élémentaires $\boldsymbol{K}_e$ et $\{f_e\}$:

$$K_e = \int_{V^e} B^T H B dV \qquad (1.62)$$

$$f_e = \int_{V_e} N^T f_v dV + \int_{S_f} N^T f_S dS_f \qquad (1.63)$$

La méthode-P présente de nombreux avantages :

- Elle est plus précise et sa convergence est plus rapide que dans les méthodes conventionnelles. En effet, pour différentes catégories de problèmes, lorsque la solution exacte est partout analytique, le taux de convergence est exponentiel, contrairement aux autres méthodes ou le taux est algébrique. Pour des problèmes dont la solution exacte contient un nombre fini de points singuliers, ce qui est le cas dans la plupart des problèmes de statique linéaire, le taux de convergence est algébrique, mais il est deux fois plus élevé que celui de la méthode-H par exemple. Aucun verrouillage numérique dû à la quasi incompressibilité de certains matériaux n'est observé.

Cependant, lorsque le coefficient de Poisson est plus proche de 0.5, la convergence asymptotique de la méthode-P s'obtient à partir d'un degré plus élevé.

- En plus du fait de la formulation hiérarchique utilisée pour la représentation des déplacements, la matrice de raideur relative à un degré donné imbrique celles de degrés inférieurs. Ceci permet d'obtenir de manière économique une séquence de solutions au lieu d'une seule solution. Les solutions convergent de manière strictement monotone, ce qui permet d'estimer l'énergie potentielle totale exacte du problème à partir de trois solutions consécutives en utilisant une procédure d'extrapolation de Richardson. L'erreur globale peut ainsi être estimée.

  La géométrie peut être représentée de manière exacte, ce qui évite les erreurs liées à sa modélisation.
- Enfin la tâche de modélisation est réduite car le maillage contient peu d'éléments et peut être directement obtenu par division des volumes en macroéléments (maillages structurés).

La méthode-P présente cependant certains inconvénients :

- la méthode-P convient bien pour une analyse détaillée des composantes d'une structure complexe. Par contre, la méthode-H semble plus adéquate pour obtenir une solution globale car le maillage structuré d'une pièce mécanique comportant plusieurs niveaux de détails est très difficile à réaliser et aboutit souvent à un grand nombre d'éléments.
- De plus, pour des problèmes dont la solution contient un nombre infini de points singuliers (cas des structures composées de plusieurs matériaux ou en régime élastoplastique.), la méthode-P n'est pas la meilleure du point de vue de la convergence, car pour un même nombre de degrés de liberté, la matrice de rigidité relative a la méthode-P est plus dense, ce qui augmente les demandes en ressources pour la résolution du système.
- Enfin la méthode-P est difficile à mettre en œuvre dans un code n'ayant pas d'outil de post-traitement pour des éléments de degré élevés.

Pour un modèle cinématiquement admissible, le champ des déplacements doit être continu. Par conséquent, les fonctions de forme doivent l'être aussi. Pour faciliter les traitements numériques, il faut choisir des fonctions relativement simples, par exemple des

polynômes. De plus, pour réduire l'erreur numérique, il faut que le système d'équations éléments finis soit bien conditionné, et ce surtout pour les problèmes à grand nombre de degrés de liberté. Ces conditions impliquent le choix des fonctions de forme construites à partir de fonctions polynomiales simples possédant certaines propriétés d'orthogonalité telles que les polynômes de Legendre. Il existe trois espaces polynomiaux bidimensionnels fréquemment utilisés :

- Espace $S^p(V)$ ou famille "Serendipity":

Il correspond à l'ensemble de tous les monômes $\xi^i \eta^j$ avec i, j = 0,1,..., p et $i+j$= 0,1,..., p. Avec en plus, si p= 1, le monôme $\xi\eta$ et, si p = 2, les monômes $\xi^p\eta$ et $\xi\eta^p$. Ces monômes supplémentaires n'augmentent pas le degré sur les côtés de l'élément de référence.

- Espace $S^{p,q}(V)$ ou famille de "Lagrange" :

Il correspond à l'ensemble des monômes $\xi^i \eta^j$ avec i= 0,1,..., p et j = 0,1,..., q.

- Espace $\tilde{S}^{p,q}(V)$ ou famille mixte :

  Il est composé de l'ensemble des monômes communs à $S^p(V)$ et à $S^{p,q}(V)$

  formellement : $\tilde{S}^{p,q}(V) = S^p(V) \cap S^{p,q}(V)$

Les fonctions de formes hiérarchiques sont généralement choisies dans l'espace de Serendipity [Szabo & Babuska 1991] ; elles sont définies à partir des polynômes de Legendre $P_i(x)$ définie [-1, 1] qui s'écrivent :

$$\begin{cases} P_0(x) = 1 \\ P_n(x) = \dfrac{1}{2^n n!}\dfrac{d^n}{dx^n}\left[\left(x^2-1\right)^n\right] \text{ , n=1,2,3,...} \end{cases} \qquad (1.64)$$

Ils sont solutions de l'équation différentielle suivante pour n = 0, 1, 2,... :

$$\left(1-2x^2\right)y'' - 2xy' + n\left(n+1\right)y = 0 \qquad (1.65)$$

Ces fonctions de forme peuvent être classées en trois catégories :

- Les fonctions de formes nodales :

$$N_i(\xi,\eta)=\frac{1}{4}(1+\xi_i\xi)(1+\eta_i\eta) \quad \text{i=1,....,4}$$

➢ Les fonctions de forme de coté :

$$\text{Coté 1 : } N_i^{(1)}(\xi,\eta)=\frac{1}{2}(1-\eta)\Phi_i(\xi) \quad \text{i = 2,....,p}$$

$$\text{Coté 2 : } N_i^{(2)}(\xi,\eta)=\frac{1}{2}(1+\xi)\Phi_i(\eta) \quad \text{i = 2,....,p}$$

$$\text{Coté 3 : } N_i^{(3)}(\xi,\eta)=\frac{1}{2}(1+\eta)\Phi_i(\xi) \quad \text{i = 2,....,p}$$

$$\text{Coté 4 : } N_i^{(4)}(\xi,\eta)=\frac{1}{2}(1-\xi)\Phi_i(\eta) \quad \text{i = 2,....,p}$$

➢ Les fonctions de forme internes :

$$N_k^0(\xi,\eta)=\Phi_i(\xi)\Phi_j(\eta) \qquad \text{k=1,....,}\frac{1}{2}(p-2)(p-3)$$

Avec i, j=2,...., p-2 ; i+j =4,...., p et $p \geq 4$

Et avec $\Phi_j(x)=\sqrt{\frac{2j-1}{2}}\int_1^x P_{j-1}(t)dt$ j=2,3,.... (1.66)

I.7 Intégration numérique

Il est clair que pour résoudre le système [K][U] = [F], il y a des intégrations à faire car pour avoir la matrice de rigidité [K] il faut intégrer sur l'ensemble de la structure. Si on utilise un ordinateur pour déterminer les solutions du système, il faut faire des intégrations numériques

I.7.1 Méthode de Newton-Cotes

Cette méthode utilise n points, x_i, répartis régulièrement sur [-1,1], avec des poids w_i. Le problème est de déterminer la position et la valeur des poids pour avoir une intégration exacte dans certain cas et donc :

$$I=\int_{-1}^{1} f(x)dx \qquad (1.67)$$

Exemple:

Prenons x_i = -1et 1. Déterminons les poids.

$$\begin{cases} w_1 + w_2 = 2 \\ (w_1 - w_2)x_1 = 0 \\ x_1^2(w_1 + w_2) = \dfrac{2}{3} \\ (w_1 - w_2)x_1^3 = 0 \end{cases} \Rightarrow \begin{cases} w_1 = w_2 = 1 \\ x_1 = -x_2 = \dfrac{1}{\sqrt{3}} \end{cases}$$

$$\Leftrightarrow \begin{cases} w_1 + w_2 = 2 \\ w_1 - w_2 = 0 \end{cases} \Rightarrow w_1 = w_2 = 1$$

Donc avec deux points -1,1 ayant des poids égaux à 1, nous intégrons de façon exacte un polynôme d'ordre 1. Mais que se passe-t-il avec un polynôme d'ordre supérieur ?

$$I = \int_{-1}^{1} ax^2 + bx + c)dx = \frac{2}{3}a + 2c$$

$$w_1 f(-1) + w_2 f(1) = 2a + 2c$$

Il semble donc qu'avec deux points et la méthode de Newton Cotes, on intègre seulement un polynôme d'ordre 1

Il est à remarquer que la méthode de Newton-Cotes intègre exactement un polynôme d'ordre n-1 avec n points.

I.7.2 Méthode de GAUSS

Le principe de la méthode reste le même, mais il faut prendre les points de façon symétrique à 0 et dans]-1, 1[.

Prenons deux points x_1, x_2, ayant des poids respectifs w_1, w_2 et un polynôme d'ordre 1

$$I = \int_{-1}^{1} (ax + b)dx = 2b = w_1 f(x_1) + w_2 f(x_2) \quad (1.68)$$

$$\Leftrightarrow \begin{cases} w_1 + w_2 = 2 \\ x_1 w_1 + x_2 w_2 = 0 \Rightarrow \text{deux types de solutions} \\ x_1 = -x_2 \end{cases}$$

$$\begin{cases} w_1 = w_2 = 1 \\ x_1 = -x_2 \end{cases} ou \quad \begin{cases} w_1 = 2 \\ x_1 = -x_2 = 0 \end{cases}$$

Donc avec un seul point il est possible de déterminer l'intégration exacte d'un polynôme d'ordre 1.

Prenons maintenant un polynôme d'ordre 3 et deux points :

$$I = \int_{-1}^{1} ax^3 + bx^2 + cx + d)dx = \frac{2}{3}b + 2d = w_1 f(x_1) + w_2 f(x_2) \qquad (1.69)$$

$$\begin{cases} w_1 + w_2 = 2 \\ (w_1 - w_2)x_1 = 0 \\ x_1^2(w_1 + w_2) = \frac{2}{3} \\ (w_1 - w_2)x_1^3 = 0 \end{cases} \Rightarrow \begin{cases} w_1 = w_2 = 1 \\ x_1 = -x_2 = \frac{1}{\sqrt{3}} \end{cases}$$

On observe qu'avec seulement deux points on intègre exactement un polynôme de degré 3

Il est à remarquer que la méthode de Gauss permet d'intégrer exactement un polynôme d'ordre 2n-1 avec n points.

La méthode de Gauss est aussi utilisée dans le cadre de la méthode-P pour l'intégration des polynômes de degré p. Les formules ainsi obtenues sont parfaitement symétriques et l'intégration utilise (p+1)*(p+1) points de Gauss pour un domaine quadrangulaire.

I.8 Modélisation des éléments surfaciques

I.8.1 Les fonctions de forme ou fonctions d'interpolation

Les fonction de forme ou fonctions d'interpolation sont les fonctions qui relient les déplacements d'un point quelconque intérieur à un élément aux déplacements nodaux qui sont les degrés de liberté dans le cas de l'approche cinématique : il y a pour un élément autant de fonctions de forme que de degrés de liberté dans l'élément .Elles assurent le passage du problème continu au problème discret, la connaissance du déplacement en quelques noeuds discrets permettant de reconstruire l'intégralité du champ de déplacement par la relation

$$u = [N]\{d\}$$

où les degrés de liberté rangés dans le vecteur $\{d\}$ sont les déplacements des noeuds du maillage. Le déplacement en un point de l'élément n'est qu'une combinaison linéaire des déplacements nodaux, dont les intensités sont les valeurs des fonctions de forme en ce point. Entre autres propriétés, la fonction de forme doit être telle que la continuité du déplacement inter-élément soit garantie.

Il en résulte que la fonction Ni associée au degré de liberté d_i y prend pour valeur 1 car le déplacement physique dans une direction en un point matériel situé sur un noeud du maillage est égal à la valeur du degré de liberté d_i qui le représente. D'autre part, elle a pour valeur 0 sur tous les autres degrés de liberté de l'élément car le déplacement sur un côté ne doit dépendre que des déplacements des noeuds situés sur ce coté pour respecter la condition de continuité inter-élément.

L'approche la plus simple pour décrire le comportement d'un élément consiste à représenter son champ de déplacement interne par des développements polynomiaux.

Dans le cas bidirectionnel par exemple, pour un élément rectangulaire à interpolation linéaire, l'approximation du déplacement sur l'élément s'écrit :

$$[U(x,y)] = \begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix}$$

Plus explicitement cette relation se met sous la forme

$$\begin{cases} u(x,y) = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 xy \\ v(x,y) = \alpha_5 + \alpha_6 x + \alpha_7 y + \alpha_8 xy \end{cases} \text{ ou } \{U_e\} = [G]\{\alpha\}$$

où $\{\alpha\}$ est le vecteur des coordonnées généralisées qui n' a aucune signification physique.

Et $$[G] = \begin{bmatrix} 1 & x & y & xy & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x & y & xy \end{bmatrix}$$

Il est nécessaire de connaître le déplacement aux nœuds de l'élément pour déterminer les inconnues α_i , i = 1,2,3,4 et pouvoir ainsi reconstruire le champ de déplacement continu : il faut si possible autant de degrés de liberté que d'inconnues généralisées. On prend les déplacements nodaux de l'élément rectangulaire par exemple :

$$\{U_e^i\} = \begin{Bmatrix} u_i \\ v_i \end{Bmatrix}, \text{ i =1, 2, 3, 4}$$

car on choisit les déplacements aux noeuds du maillage pour inconnues du problème.
En considérant les coordonnées des nœuds d'un élément la relation (xx) permet d'écrire :

$$\{U_e\} = [C]\{\alpha\}$$

où $\{\alpha\}$ et un vecteur colonne constitué des huit coefficients $\alpha_{i,i=1,2,..8}$ et

$[C]$ est la matrice des coordonnées nodales

$$[C]=\begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_2 & y_2 & x_2y_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x_2 & y_2 & x_2y_2 \\ 1 & x_3 & y_3 & x_3y_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x_3 & y_3 & x_3y_3 \\ 1 & x_4 & y_4 & x_4y_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x_4 & y_4 & x_4y_4 \end{bmatrix}$$

Lorsque la matrice $[C]$ est inversible, il est possible d'exprimer les coordonnées, généralisées en fonction des degrés de liberté :

$$[\alpha]=[C^{-1}][U_e] \quad (5.3)$$

soit encore $$[U]=[G][C^{-1}][U_e]=[N_e][U_e] \quad (5.4)$$

La matrice $[N_e]$ est la matrice des fonctions d'interpolation ou fonctions de forme de l'élément, elle contient les fonctions N_i associées à chaque degré de liberté.

Elle est donnée par :

$$[N_e]=\begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix}$$

et les fonctions d'interpolation N_i , i=1,2,3 ,4 sont données en fonction des coordonnées paramétriques par :

$$N_1 := \frac{1}{4}(\eta-1)(\xi-1) \qquad N_2 := -\frac{1}{4}(\eta-1)(1+\xi)$$

$$N_3 := \frac{1}{4}(1+\eta)(1+\xi) \qquad N_4 := -\frac{1}{4}(1+\eta)(\xi-1)$$

Si $[C]$ n'est pas inversible, c'est qu'il n'y a pas unicité de la relation entre les coordonnées généralisées et les degrés de liberté. Il existe alors différentes techniques permettant néanmoins d'inverser $[C]$ de manière à définir les fonctions de forme, comme par exemple la condensation de certains degrés de liberté. Le champ de déplacement devant être continu dans la structure, les fonctions de forme le sont aussi.

Plus généralement, si le déplacement est de classe C^i, alors les fonctions de forme doivent l'être aussi. De manière a pouvoir représenter des mouvements de corps rigides ou des états de contrainte localement ou globalement uniformes, les développements polynomiaux initiaux doivent contenir un terme constant non nul et les termes linéaires.

Pour un élément triangulaire du premier degré, comportant trois degrés de liberté de translation sur x et sur y, il faut un terme constant et deux termes linéaires, définissant un polynôme complet d'ordre 1:

$$u(x,y) = \alpha_0 + \alpha_1 x + \alpha_2 y$$

Pour un triangle du second degré, il faut trois termes de plus. On peut choisir x^2, y^2 et xy ce qui conduit à un polynôme complet d'ordre deux.

I.8.2 Fonction de forme type Lagrange

Une façon simple de générer des fonctions de forme pour les éléments rectangulaires ou cubiques orientés parallèlement aux axes structuraux consiste à effectuer des produits de polynômes de chaque variable, la continuité le long des bords devant être respectée d'un élément à celui qui lui est adjacent. Cette condition de continuité impose qu'il y ait exactement (n+1) valeurs connues sur les bords d'un élément de degré n, donc n noeuds. En supposant une approximation linéaire selon chaque axe, on détermine

$$u(x,y) = (a_0 + a_1 x)(b_0 + b_1 y) \quad \text{ou} \quad u(x,y,z) = (a_0 + a_1 x)(b_0 + b_1 y)(c_0 + c_1 z)$$

Ce même type de développement peut être réalisé pour des interpolations paraboliques, cubiques ou plus selon chaque direction. Pour des degrés supérieurs à 1 , il est nécessaire de

définir des noeuds sur chaque arête mais aussi des nœuds internes, ce qui enrichit le comportement de l'élément mais augmente le nombre de degré de liberté de 1'élément (figure 5.1).

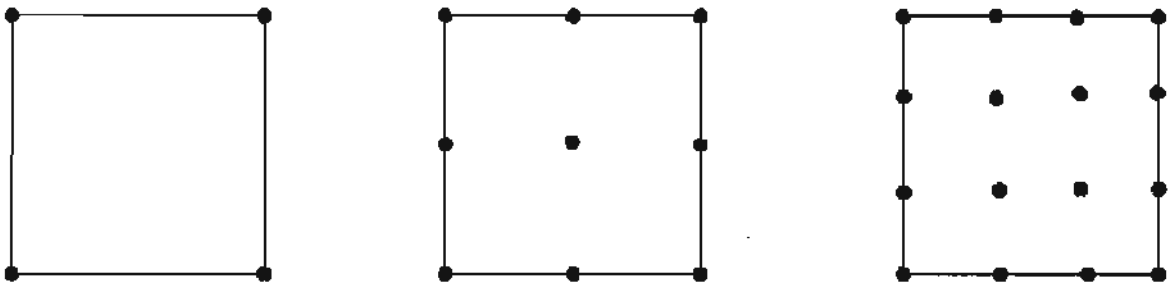


Fig.I.1 : Position des nœuds pour les carrés de Lagrange

Les fonctions de forme associées à ces noeuds internes, appelées « modes bulles », n'ont pas d'incidence sur les éléments voisins. C'est la raison pour laquelle on élimine parfois des degrés de liberté internes par condensation, le temps nécessaire pour cette opération au niveau de chaque élément étant nettement compensé par le gain de temps résultant de la réduction de taille du système à résoudre.

Tous calculs faits, les fonctions de forme N_i sont des produits pour chaque direction des polynômes de Lagrange d'ou le nom donné à cette famille. On peut aussi les construire indirectement par application de la formule les définissant.

Pour une barre du premier degré parallèle à l'axe des x, avec $x_1 = 0$ et $x_2 = L$,

$$N_1(x) = \frac{x - x_2}{x_1 - x_2} = 1 - \frac{x}{L} \quad \text{et} \quad N_2(x) = \frac{x - x_1}{x_1 - x_2} = \frac{x}{L}$$

Au second degré, on calcule directement avec $x_1 = 0$, $x_2 = L/2$ et $x_3 = L$:

$$N_1(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} = (2\frac{x}{L} - 1)(\frac{x}{L} - 1)$$

$$N_2(x) = 4\frac{x}{L}(1 - \frac{x}{L})$$

$$N_3(x) = \frac{x}{L}(2\frac{x}{L} - 1)$$

Même lorsqu'il s'agit de fonctions de forme associées à des degrés de liberté dans le plan de l'élément, la représentation graphique qui en est faite est transversale, ce n'est que la valeur prise par la fonction en un point.

La figure 5.3 illustre la fonction de forme associée à un nœud d'élément quadrangulaire linéaire : elle prend la valeur 1 sur le noeud considéré et 0 sur les autres pour garantir la condition de continuité inter-éléments.

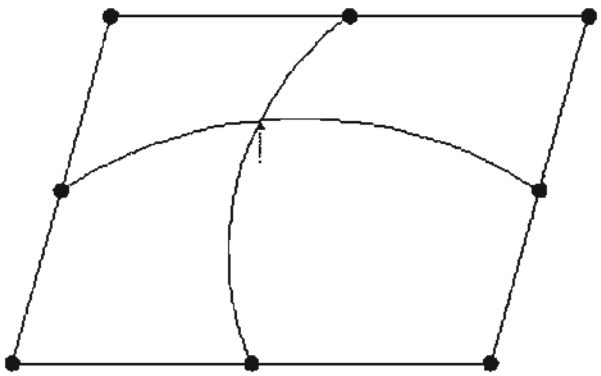


Fig.I.2 Mode bulle associée à un nœud interne

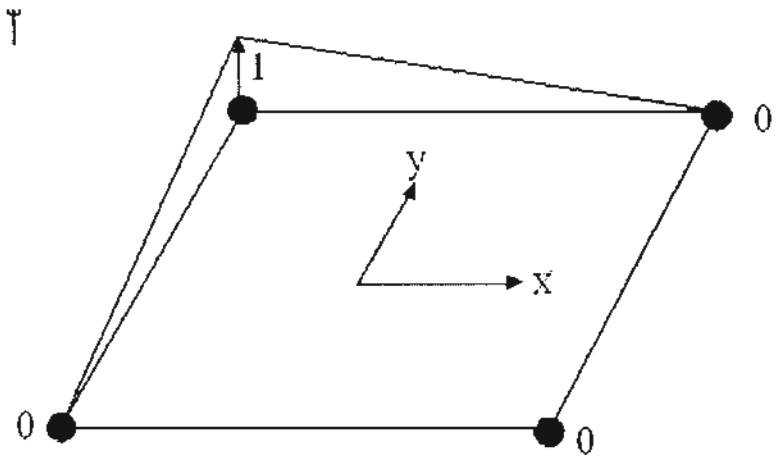


Fig.I.3 : Fonction de forme associée à un nœud sommet

I.8.3 Fonction de forme type Serendip

On ne rajoute des noeuds que sur les arêtes des éléments pour augmenter leur degré. Au premier degré, les éléments de Lagrange et de Serendip sont identiques.

Au second degré, le rectangle de Serendip est défini par 8 noeuds alors que celui de Lagrange est défini par 9 noeuds (figure 5.4). Les fonctions de forme de ces éléments sont classiques et connues.

Au second degré par exemple, pour des carrés de coté 2 dont les bords sont parallèles aux axes structuraux, les coordonnées x et y variant de -1 à +1, les fonctions de forme sont :

pour le noeud x = 1 et y =1 $N_i = -1/4(1+x)(1+y)(1-x-y)$

pour le nœud milieu x = 0 et y =1 $N_i = -1/2(1-x^2)(1+y)$

pour le nœud milieu y =0 et x = -1 $N_i = 1/2(1-y^2)(1+x)$

Pour déterminer les fonctions manquantes, il suffit de remplacer x par -x et y par -y.
Ces fonctions de forme sont faciles à construire pour les deux familles si les éléments ont des bords droits parallèles aux axes structuraux, si les quadrilatères sont des rectangles ou des carrés, les hexaèdres des cubes. Dans le cas d'un quadrilatère non rectangulaire (figure xx), se pose le problème suivant l'expression du champ de déplacement fait intervenir un terme en x^2 .Or il n'y a que deux noeuds sur ce cote, donc la continuité avec l'élément voisin ne pourra pas être garantie.

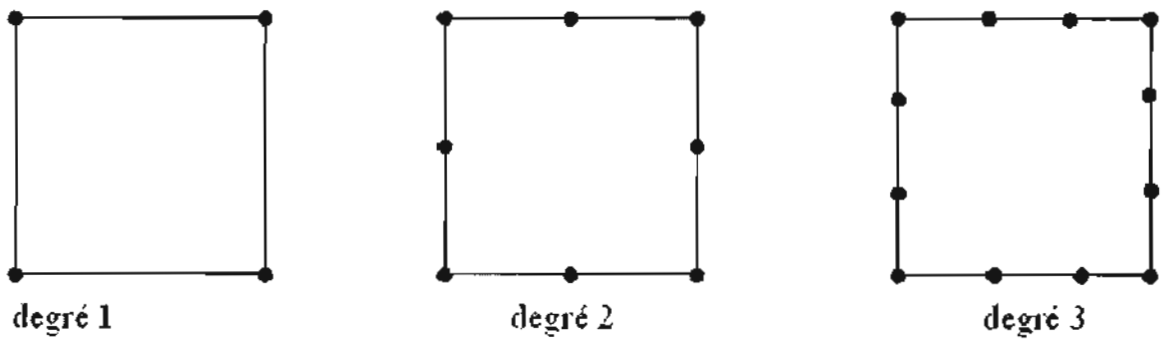


Fig. I.4: Position des noeuds pour les carrés de Serendip

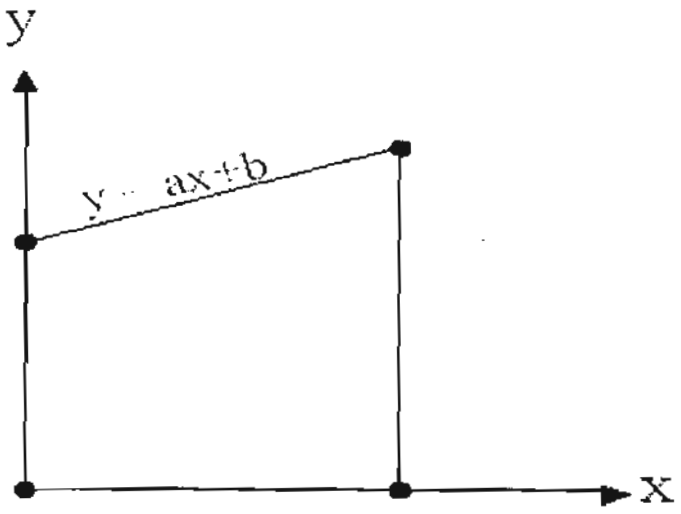


Fig. I.5: Quadrilatère non rectangulaire

La détermination des fonctions de forme d'un élément de forme simple mais dont les sommets et les noeuds milieux éventuels ont des coordonnées (x, y) ou (x, y, z) quelconques est quasiment impossible analytiquement. Les fonctions de forme sont par contre relativement faciles à déterminer pour des éléments très particuliers tels le carré de coté 2, le cube de coté 2, dont les bords sont parallèles aux axes structuraux ou dans le repère barycentrique pour les éléments basés sur une formulation triangulaire.

Ces éléments de forme particulière, décrits dans le système de coordonnées intrinsèques (ξ,η) ou (ξ,η,ζ) , ces paramètres variant de -1 a + 1, sont appelés éléments de référence ou éléments parents et permettent d'écrire simplement les conditions de continuité sur les bords. Ils jouent un très grand rôle en pratique car pour évaluer les caractéristiques de raideur de la plupart des éléments vrais, on se ramène aux éléments parents qui leur correspondent.

I.8.4 Eléments de membrane

Une membrane est une structure plane dont une dimension est très petite par rapport aux deux autres et qui n'est sollicitée que par des charges dans son plan lorsqu'elle est bidimensionnelle. Le s membranes peuvent en effet être soit bidimensionnelles soit tridimensionnelles, mais dans le repère propre attaché à chaque élément, le comportement mécanique est de l'état plan de contrainte. Par définition, les membranes n'ont pas de raideur transversale : dans un certain nombre de cas, les programmes introduisent des axes locaux

automatiques, détectent des pivots nuls ou presque nuls et bloquent des degrés de liberté dans ce système local en dénaturant le problème posé. Pour cette raison essentielle, on préfère utiliser des coques dans l'espace et on réserve les membranes pour des modèles géométriquement bidimensionnels, ou pour le calcul des contraintes de peau en tapissant les volumes de membranes

L'épaisseur e est toujours dirigée selon l'axe z. Le plan de symétrie situé à mi-épaisseur est conventionnellement le plan xOy et porte le nom de feuillet moyen (fig. I.5).

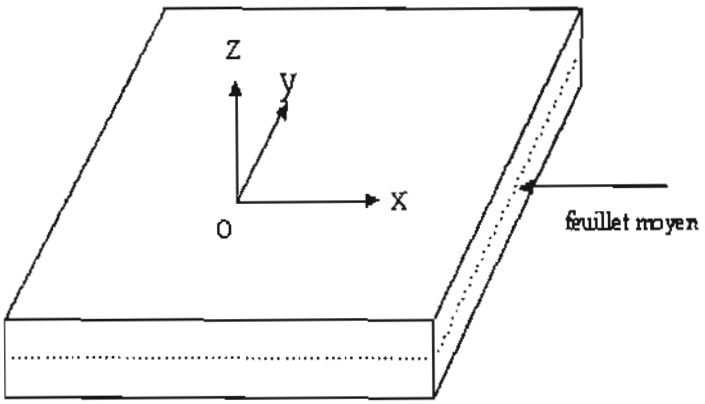


Fig.I.6 : feuillet moyen et axes d'une membrane

Hypothèses de la théorie des membranes :

- Il n'y a aucune charge transversale sur la membrane $\sigma_{zz} = 0$;
- Aucun effort tranchant n'est appliqué perpendiculairement au plan de la membrane $\tau_{xz} = 0$, par réciprocité les contraintes tangentielles τ_{zx} et τ_{zy} sont nulles;
- Aucun moment ne peut être appliqué autour d'un axe quelconque du plan, car cela engendrerait des contraintes τ_{xz} et τ_{zy} et un déplacement transversal perpendiculaire au feuillet moyen,

La déformation ε_{zz} est non nulle et est donnée par la relation suivante :

$$\varepsilon_{zz} = -\frac{\nu}{E}\left(\sigma_{xx} + \sigma_{yy}\right)$$

Elle correspond à la variation relative d'épaisseur de la membrane sous la charge appliquée. Une caractéristique du comportement membranaire est la symétrie du champ de contrainte par rapport au feuillet moyen: sous l'action d'une charge dans le plan, la contrainte est uniformément répartie dans l'épaisseur (fig.I.7).

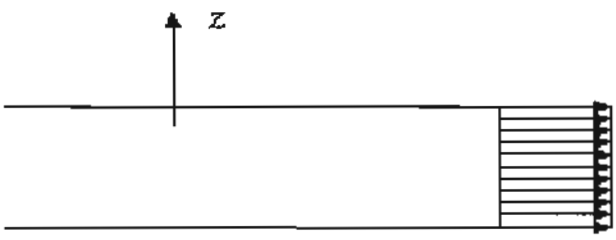


Fig.I.7 : Répartition des contraintes dans une membrane

On définit les flux d'efforts dans la membrane par les trois relations suivantes :

$$N_x = \int_{-e/2}^{+e/2} \sigma_{xx} dz \quad N_y = \int_{-e/2}^{+e/2} \sigma_{yy} dz \quad N_{xy} = \int_{-e/2}^{+e/2} \tau_{xy} dz$$

Ce sont des forces par unité de longueur qui s'expriment en Newton par mètre (N/m).
La membrane est un élément de l'élasticité bidimensionnelle à définition surfacique, seul est décrit son feuillet moyen. Il est donc indispensable de donner au programme l'épaisseur de l'élément, répartie symétriquement de part et d'autre du feuillet moyen. Différents types de mailleurs automatiques bidimensionnels permettent la génération de ces éléments, de forme triangulaire ou quadrangulaire.
Outre la topologie, il faut obligatoirement fournir à un programme d'analyse linéaire pour un matériau isotrope les caractéristiques suivantes:

- Module de Young,
- Coefficient de Poisson,
- Epaisseur.

Masse volumique, coefficient de dilatation thermique ne sont nécessaires que pour Certains type de calcul : analyse modale, réponse dynamique, calcul statique sous chargement thermique.

I.8.5 Eléments finis de membrane

Il existe généralement deux éléments de membranes, un triangulaire et un rectangulaire, définis conventionnellement ans le plan xOy (fig.I.5).

Au premier degré, ces éléments ont des bords géométriquement rectilignes .Chaque nœud possède deux degrés de liberté de translation dans le plan, notés u et v tous les deux alimentés raideur. Le triangle est topologiquement défini par 3 noeuds et a 6 degrés de liberté. Le quadrangle est topologiquement défini par 4 nœuds et a 8 degrés de liberté. Ces éléments sont en général sensiblement trop raides et il en faut un grand nombre pour converger vers la solution attendue. Il existe néanmoins dans certains codes des éléments < extérieurement> du premier degré, à bords rectilignes, mais qui, de par leur construction (intégration sélective du cisaillement) donnent d'excellents résultats.

Les membranes sont souvent disponibles au second degré. Le degré des champs de déplacement est enrichi par l'adjonction d'un noeud supplémentaire sur chaque bord. C'est le noeud milieu qui possède lui aussi les deux degrés de liberté de translation dans le plan u et v. Il devient alors possible de prendre en compte la courbure géométrique de l'élément. Le triangle est défini par 6 noeuds et 12 degrés de liberté, le quadrangle par 8 noeuds et 16 degrés de liberté. Ces éléments sont plus souples que ceux du premier degré et le raffinement du maillage peut parfois sembler grossier bien que la solution soit numériquement satisfaisante

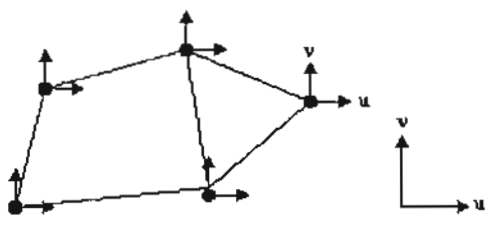


Fig.I.8 : éléments du premier degré

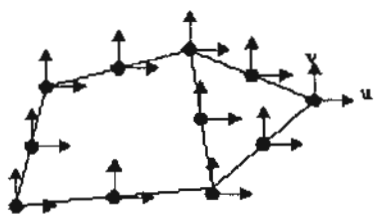


Fig.I.9 : éléments du second degré

Dans le cas de membranes tridimensionnelles dans le repère propre de l'élément, un noeud n'a que deux degrés de liberté mais dans l repère structural, il y en a trois par noeud dont un qui peut être non alimenté en raideur selon l'orientation de la membrane par rapport au repère.

I.8.6 Eléments de plaque

Une plaque est un solide limité par deux plans parallèles voisins d'équation $\mathbf{z} = \pm\frac{1}{2}\mathbf{t}$, où t(x,y) est l'épaisseur de la plaque. Le plan (Oxy) est appelé plan moyen (ou feuillet moyen) de la plaque et aussi plan de référence.

Ce sont des structures en état plan de contraintes admettant des déplacements verticaux suivant l'axe z.

Hypothèses de la théorie des plaques

- Le milieu est homogène et isotrope.
- La déformation transversale ε_{zz} est nulle ($\varepsilon_{zz} = 0$)
- La contrainte σ_{zz} est négligée par rapport aux autres composantes du tenseur des contraintes.
- La prise en compte de la variation du cisaillement transversal par l'introduction d'un coefficient correctif k sur les contraintes tangentielles.
- Les inconnues cinématiques utilisées seront : le déplacement transversale w et les rotations θ_x et θ_y des sections normales au plan moyen dans les plans (Oxz) et (Oyz) respectivement.
- Les efforts de membrane étant négligés on aura les moments de flexion M_x, M_y, M_{xy} et les efforts tranchants Q_x et Q_y produit par le chargement de la plaque sur son plan moyen.

I.8.6.1 Equations cinématiques

Ces hypothèses nous permettent d'établir que les déformations linéaires s'écrivent :

$$\boldsymbol{\varepsilon} = e - z\boldsymbol{\chi}$$

avec $\boldsymbol{\varepsilon}^T = \left[\varepsilon_x \quad \varepsilon_y \quad \varepsilon_{xy} \quad \gamma_{xz} \quad \gamma_{yz}\right]$;

Déformations de membrane : $\mathbf{e}^T = \begin{bmatrix} \mathbf{u}_x & \mathbf{u}_y & \mathbf{u}_x + \mathbf{u}_y \end{bmatrix}$;

Courbures : $\chi^T = \begin{bmatrix} \beta_{x,x} & \beta_{y,y} & \beta_{x,y} + \beta_{y,x} \end{bmatrix}$;

Déformations de cisaillement transversal : $\gamma = \begin{bmatrix} \mathbf{w}_{,x} + \beta_x & \mathbf{w}_{,y} + \beta_y \end{bmatrix}$

Pour l'effet flexionnel, le champ de déformation s'écrit :

$$\varepsilon = \begin{bmatrix} \chi \\ \gamma \end{bmatrix} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{bmatrix} = \begin{bmatrix} -\mathbf{z}\beta_{x,x} \\ -\mathbf{z}\beta_{y,y} \\ -\mathbf{z}\left(\beta_{x,y} + \beta_{y,x}\right) \\ \mathbf{w}_{,y} + \beta_y \\ \mathbf{w}_{,x} + \beta_x \end{bmatrix}$$

Ou encore :

$$\varepsilon = LU$$

$$\begin{bmatrix} \beta_{x,x} \\ \beta_{y,y} \\ \left(\beta_{x,y} + \beta_{y,x}\right) \\ \mathbf{w}_{,x} + \beta_x \\ \mathbf{w}_{,y} + \beta_y \end{bmatrix} = \begin{bmatrix} \partial/\partial\mathbf{x} & 0 & 0 \\ 0 & \partial/\partial\mathbf{y} & 0 \\ \partial/\partial\mathbf{y} & \partial/\partial\mathbf{x} & 0 \\ 1 & 0 & \partial/\partial\mathbf{x} \\ 0 & 1 & \partial/\partial\mathbf{y} \end{bmatrix} \begin{bmatrix} \beta_x \\ \beta_y \\ \mathbf{w} \end{bmatrix}$$

Avec :

$$\beta_x = -\theta_y$$

$$\beta_y = -\theta_x$$

$$\begin{bmatrix} -\theta_{y,x} \\ \theta_{x,y} \\ \left(\theta_{x,x} - \theta_{y,y}\right) \\ w_x - \theta_y \\ w_x + \theta_x \end{bmatrix} = \begin{bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial y & 0 \\ \partial/\partial y & \partial/\partial x & 0 \\ 1 & 0 & \partial/\partial x \\ 0 & 1 & \partial/\partial y \end{bmatrix} \begin{bmatrix} -\theta_y \\ \theta_x \\ w \end{bmatrix}$$

I.8.6.2 Equations d'équilibre

Les sollicitations sur les plaques créent :

- des contraintes Normales : σ_x σ_y
- des contraintes de cisaillement : τ_{xy} τ_{yz} τ_{xz} (fig.I.9)

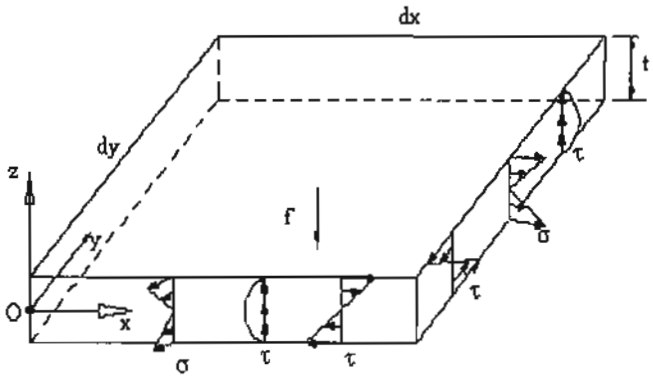


Fig.I.10 : répartition des contraintes dans une plaque

Cet état de contraintes engendre :

- les moments : M_x M_y M_{xy}
- les efforts de cisaillement ou efforts tranchants : Q_x et Q_y (fig.I.10)

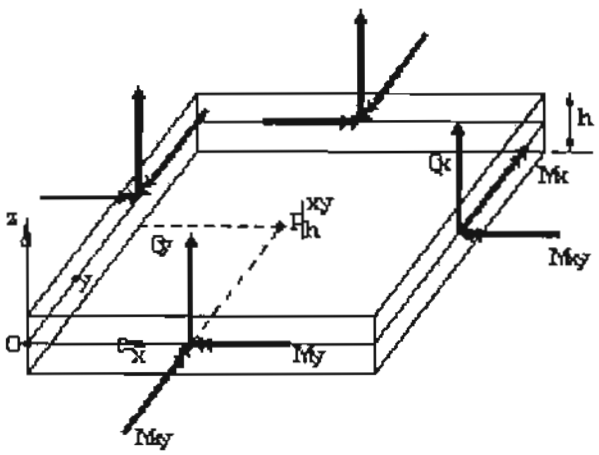


Fig. I.11: efforts résultant sur une plaque

Les équations d'équilibre en un point P(x,y,z) sont :

$$\sigma_{x,x} + \sigma_{xy,y} + \tau_{xz,z} + \mathbf{f}_{vx} = 0$$

$$\sigma_{xy,x} + \sigma_{y,y} + \tau_{yz,z} + \mathbf{f}_{vy} = 0$$

$$\tau_{xz,x} + \tau_{yz,y} + \sigma_{z,z} + \mathbf{f}_{vz} = 0$$

f_{vz}, f_{vy}, f_{vx} sont des forces par unité de volume.

En faisant l'équilibre global sur l'épaisseur on obtient :

$$\mathbf{Q}_{x,x} + \mathbf{Q}_{y,y} + \mathbf{f}_z = 0 \qquad (1.10)$$

L'équilibre des moments par rapports aux axes x et y s'écrit :

$$\mathbf{M}_{x,x} + \mathbf{M}_{xy,y} + \mathbf{Q}_x + \mathbf{m}_x = \mathbf{0}$$

$$\mathbf{M}_{xy,x} + \mathbf{M}_{y,y} + \mathbf{Q}_y + \mathbf{m}_y = 0 \qquad (1.11)$$

Avec :

$$F = \begin{Bmatrix} f_x \\ f_y \\ f_z \end{Bmatrix} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \begin{Bmatrix} f_{vx}(x,y,z) \\ f_{vy}(x,y,z) \\ f_{vz}(x,y,z) \end{Bmatrix} dz \quad ;$$

$$m = \begin{Bmatrix} m_x \\ m_y \end{Bmatrix} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \begin{Bmatrix} f_{vx}(x,y,z) \\ f_{vy}(x,y,z) \end{Bmatrix} z dz$$

$\mathbf{f}_x, \mathbf{f}_y, \mathbf{f}_z, \mathbf{m}_x, \mathbf{m}_y$ sont des efforts par unité de surface moyenne

On a les moments et efforts tranchants qui sont données par :

$$\mathbf{M}_x = \int_{-t/2}^{t/2} \sigma_x z dz, \quad \mathbf{M}_y = \int_{-t/2}^{t/2} \sigma_y z dz, \quad \mathbf{M}_{xy} = \int_{-t/2}^{t/2} \sigma_{xy} z dz \qquad (1.12)$$

$$\mathbf{Q}_x = \int_{-t/2}^{t/2} \tau_{zx} dz, \quad \mathbf{Q}_y = \int_{-t/2}^{t/2} \tau_{zy} dz \qquad (1.13)$$

M_x , M_y et M_{xy} : moments ou efforts résultants de flexion (N.m/m)

Q_x et Q_y : efforts résultants de cisaillement ou efforts tranchants (N/m)

$$\begin{Bmatrix} \mathbf{M}_x \\ \mathbf{M}_y \\ \mathbf{M}_{xy} \end{Bmatrix} = \int_{-t/2}^{t/2} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{Bmatrix} \mathbf{z}\mathbf{dz} \quad ; \quad \begin{Bmatrix} \mathbf{Q}_x \\ \mathbf{Q}_y \end{Bmatrix} = \int_{-t/2}^{t/2} \begin{Bmatrix} \tau_{zx} \\ \tau_{zy} \end{Bmatrix} \mathbf{dz} \qquad (1.14)$$

En un point situé sur le contour S on a les efforts tranchants Q_n et moments M_{xn}, M_{yn} qui s'écrivent :

$$\begin{aligned} \mathbf{Q}_n &= \mathbf{Q}_x \mathbf{n}_x + \mathbf{Q}_y \mathbf{n}_y \\ \mathbf{M}_{xn} &= \mathbf{M}_x \mathbf{n}_x + \mathbf{M}_{xy} \mathbf{n}_y \\ \mathbf{M}_{yn} &= \mathbf{M}_{xy} \mathbf{n}_x + \mathbf{M}_y \mathbf{n}_y \end{aligned} \qquad (1.15)$$

Où n_x et n_y représentent les cosinus directeurs de la normale en S.

Sur un contour S_f partie du contour A où les efforts sont imposés, les Q_n, M_{xn}, M_{yn} représentent les efforts imposés notés F_z, M_x, M_y.

Sur un contour S_u partie de A où les rotations et déplacements sont imposés les Q_n, M_{xn}, M_{yn} représentent les réactions d'appui.

Ces relations d'équilibre peuvent s'écrire sous forme matricielle par :

$$L^T \sigma + b = 0 \; ; \qquad (1.16)$$

$$\begin{bmatrix} \partial/\partial x & 0 & \partial/\partial y & 1 & 0 \\ 0 & \partial/\partial y & \partial/\partial x & 0 & 1 \\ 0 & 0 & 0 & \partial/\partial x & \partial/\partial y \end{bmatrix} \cdot \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \\ Q_X \\ Q_Y \end{Bmatrix} + \begin{Bmatrix} m_x \\ m_y \\ f_z \end{Bmatrix} = \{0\} \qquad (1.17)$$

I.8.7 Eléments de coque

Une coque est un solide ou partie de solide dont une dimension appelée épaisseur est petite vis-à-vis des deux autres et qui admet un plan de symétrie passant par le milieu de l' épaisseur, appelée feuillet moyen.

Par convention, dans le repère de l'élément, ce plan de symétrie est le plan xOy et l' axe z lui est perpendiculaire.

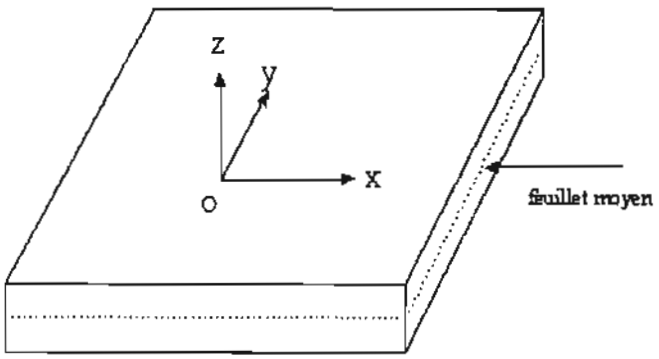


Fig.I.12 : Feuillet moyen d'une coque

Dans ce chapitre, nous n'étudierons que les coques à définition surfacique, c'est-à-dire représentées uniquement par leur feuillet moyen. Les coques épaisses, à définition volumique, c'est-à-dire explicitement définies par leur topologie tridimensionnelle, s'apparentent aux éléments de volume et ne sont pas étudiées. Une coque est la superposition d'une membrane (ne travaillant que dans son plan) et d'une plaque (ne travaillant que perpendiculairement à son plan). C'est un élément très général puisqu'il accepte tous les types de chargement : la coque est l'element fini le plus largement utilise. il peut y avoir couplage entre les deux effets mécaniques membrane/plaque, mais en règle générale, pour la modélisation des structures constituées de matériaux isotropes en analyse linéaire, les comportements membranaires et flexionnels sont découplés tant que les coques sont planes et sont étudiés séparément.

Les contraintes de plaque sont reparties antisymétriquement par rapport au feuillet moyen, distinguant ainsi les problèmes de flexion de ceux d'extension pour lesquels la répartition des contraintes est symétrique par rapport au feuillet moyen (figure...).

Fig. I.13: contraintes dans une membrane et une plaque

Les coques sont triangulaires ou quadrangulaires, à bords rectilignes ou courbes elles peuvent être générées par des mailleurs automatiques bidimensionnels ou tridimensionnels surfaciques. Outre la topologie, il faut obligatoirement fournir à un programme d'analyse linéaire, pour une structure en matériau isotrope, les caractéristiques suivantes :

- module de Young,
- coefficient de Poisson,
- épaisseur.

Masse volumique, coefficient de dilatation thermique, ne sont nécessaires que pour certains types de calcul.

I.8.7.1 Hypothèses de la théorie des coques

Dans la théorie linéaire des plaques isotropes, trois ou quatre hypothèses sont généralement admises [réf.1]:

Hypothèse 1 : les contraintes normales au plan de la plaque σ_{zz} sont négligeables par rapport aux contraintes existant dans le plan de la plaque : $\sigma_{zz} = 0$

Hypothèse 2 : les pentes du feuillet moyen restent petites par rapport à l'unité, et ceci dans n'importe quelle direction. De plus, le feuillet moyen ne subit aucune déformation du fait de la flexion.

Hypothèse 3 : les points situés dans un plan orthogonal au feuillet moyen sont dans un plan après déformation.

Hypothèse 4 : un plan orthogonal au feuillet moyen avant déformation l'est encore après déformation.

Les hypothèses 1-2-3-4 conduisent aux coques dites minces, dites de Kirchhoff dites sans déformation à l'effort tranchant.

Les hypothèses 1-2-3 conduisent aux coques d'épaisseur modérée, dites de Mindlin ou Reissner ou Hencky, dites avec déformation à l'effort tranchant.

La théorie basée sur les hypothèses 1 à 4 donne de bons résultats tant que l'épaisseur reste petite par rapport aux autres dimensions du panneau ou de la structure, et lorsque le matériau possède un module de cisaillement transversal G élevé. Pour un matériau isotrope, G est du même ordre de grandeur que le module de Young E et l'élancement géométrique seul permet de choisir le type de coque adaptée à la structure à modéliser. L'élancement à prendre en compte pour les
coques à définition surfacique est celui de la structure ou du panneau et non pas celui de l'élément fini : un panneau de 20x20x1 est considère comme mince (théorie de Kirchhoff), et peut être modélisé par 20x20 éléments de coque mince, même si chaque élément pris individuellement est un cube de1x1x1.
Dans certains cas, il est nécessaire d'apporter à cette théorie une amélioration en supprimant l'hypothèse 4 : la rotation de la section est alors différente de la pente du feuillet moyen.

I.9 Procédure de résolution

La méthode des éléments finis est une technique particulière d'approximation de fonctions solutions par sous-domaines. La forme variationnelle définie sur le milieu continu est ainsi représentée par une forme variationnelle dite discrétisée qui fait intervenir les inconnues nodales {U} qui sont les valeurs des fonctions d'approximation en certains points appelés nœuds de chaque sous-domaine.

I.9.1Démarche de résolution par éléments finis

Apres avoir posé le problème physique à résoudre sous forme d'une équation différentielle ou aux dérivées partielles à satisfaire en tout point du domaine Ω, avec des conditions aux limites sur le bord $\partial\Omega$; on construit une formulation intégrale du système différentiel à résoudre et de ses conditions aux limites : c'est la formulation variationnelle du problème. A partir de là nous procédons comme suit :

- On divise Ω en sous domaines : c'est le maillage. Les sous-domaines appelés mailles ont un volume V^e tel que :

$V = \sum v^e$ = volume totale du domaine

Et $W = \sum W^e$

- On choisit la famille de champs locaux, c'est-à-dire à la fois la position des nœuds dans les sous-domaines et les polynômes (ou autres fonctions) qui définissent le champ local en fonction des valeurs aux nœuds. La maille complétée par ces informations et ainsi appelée élément. Ainsi :

$$\{x(\xi)\} = [N(\xi)]\{x_n\} \qquad (1.70)$$

Où : $\{x\}$ position d'un point ; $\{x_n\}$ coordonnées des nœuds définissant V^e

ξ Coordonnées paramétriques $\xi,\ \eta,\ \zeta$

[N] fonctions d'interpolation en variables paramétriques

La représentation de la fonction solution {u} sur chaque élément :

$$\{u(\xi)\} = [N(\xi)]\{u_n\} \quad ; \quad \{u^*(\xi)\} = [N(\xi)]\{u^*_n\} \qquad (1.71)$$

{u} fonctions solutions : {u*} fonctions virtuelles

$\{u_n\}$ variables nodales caractérisant la fonction solution

- On ramène le problème à un problème discret : c'est la discrétisation. En effet, toute solution approchée est complètement déterminée par les valeurs aux nœuds des éléments. Il suffit donc de trouver les valeurs à attribuer aux nœuds pour décrire une solution approchée. On procède au calcul élémentaire : sur chaque élément on calcule W^e tel que $W = \sum W^e$ on a :

$$W^e = \langle u_n^* \rangle \left([k]\{u_n\} - \{f_n\} \right) \qquad (1.72)$$

[k] matrice de rigidité de l'élément et f_n vecteur élémentaire des sollicitations.

on procède ensuite à l'assemblage des matrices de rigidité et des vecteurs des sollicitations de chaque élément pour obtenir une matrice globale notée **[K]** matrice de rigidité de tout le système et un vecteur **{F}** représentant toutes les sollicitations que subit le système.

$$W = \sum_e W^e = \sum_e \langle u_n^* \rangle \left([k]\{u_n\} - \{f_n\} \right) \quad (1.73)$$

$$W = \langle U^* \rangle \left([K]\{U\} - \{F\} \right) = 0 \quad \forall \{U^*\} \quad (1.74)$$

soit $$[K]\{U\} = \{F\}$$

- Résolution, en tenant compte des conditions aux limites, trouver **{U}** tel que :

$$[K]\{U\} - \{F\} = 0$$

Dans le cas où le système est linéaire c'est-à-dire si la matrice **[K]** est définie positive, nous obtenons :

$$\{U\} = [K]^{-1}\{F\}$$

Nous obtenons ainsi le vecteur **{U}** représentant le champ de déplacements qui est l'inconnue du problème.

I.9.2 Post-traitement

Afin de permettre à l'utilisateur une analyse aisée de la solution éléments finis on procède à l'évaluation de certaines grandeurs qui dépendent fortement de la physique du problème étudié, dans le cas de l'élasticité linéaire, ces grandeurs sont les déformations et les contraintes. À partir du vecteur solution du système d'équations linéaires à l'aide des relations de compatibilité (pour le calcul des déformations) et de la loi de Hooke généralisée (pour les contraintes) c'est-à-dire :

$$\varepsilon_x = \frac{\partial u}{\partial x} \; ; \; \varepsilon_y = \frac{\partial v}{\partial y} \; ; \varepsilon_z = \frac{\partial w}{\partial z}$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \quad \gamma_{zy} = \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \quad \gamma_{xz} = \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}$$

et

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ 2\varepsilon_{yz} \\ 2\varepsilon_{xz} \\ 2\varepsilon_{xy} \end{bmatrix}$$

Partie II : Interface graphique et code de calcul par éléments finis

On a connu au cours des dernières décennies un avancement remarquable dans le domaine du calcul numérique et en mécanique des structures. Cette avancée est surtout due au développement dans la conception des ordinateurs conjugué au développement parallèle en génie logiciel. C'est ainsi que les logiciels de calculs scientifiques empruntant de plus en plus la voie des langages naturels ont connu un essor remarquable. Des outils numériques de plus en plus conviviaux et performants permettent de résoudre des problèmes réels de plus en plus complexes dans divers domaines comme le génie civil, le génie mécanique, la physique, les mathématiques appliquées, la géologie, l' astrophysique, la géophysique, etc. - .

Dans cette seconde partie du projet, nous proposons de développer et d'appliquer la méthode des éléments finis aux problèmes d'analyse structurale des éléments surfaciques dans un environnement éléments finis orienté objet et de pré développer un outil de calcul numérique pouvant être réalisé avec Visual C++.

I. Rappel de la programmation orientée objet POO

D'abord pourquoi le C++ :c'est avant tout une nécessité de répondre à des besoins générés par de gros projets. Il nécessite une façon de travailler plus rigoureuse pour un code plus structuré, extensible, réutilisable et enfin si possible portable. Ceci est assez limité lorsqu'on utilise un langage simplement structuré tel que le C ou Turbo Pascal.

La programmation orientée objet (POO)

La POO est une solution qui permet d'introduire le concept objet justement, qui consiste en un ensemble de données et de procédures qui agissent sur ces données.

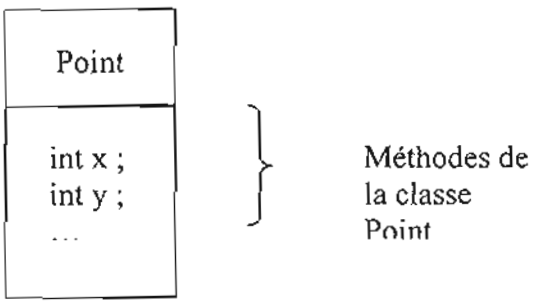
Lorsque l'objet est parfaitement bien écrit ,il introduit la notion d'encapsulation des données Ceci signifie qu'il n'est plus possible pour l'utilisateur de l'objet d'accéder directement aux données : il doit passer par des méthodes spécifiques écrites par le concepteur de l'objet et qui servent d'interface entre l'objet et ses utilisateurs .l'intérêt de cette technique est évidente :l'utilisateur ne peut pas intervenir directement sur l'objet ,ce qui diminue les risques d'erreur,ce dernier devenant une boite noire .

Une autre notion importante en POO est la notion d'héritage. elle permet la définition d'une classe nouvelle à partir d'une classe existante .il est alors possible de lui adjoindre de nouvelles données ,de nouvelles fonctions membres (procédures) pour la spécialiser .

I. 1 Notion de classe :

Une classe est une structure complexe qui permet l'encapsulation de données .elle est composée de données et de méthodes. Lorsque l'encapsulation de données est parfaite, seules certaines méthodes sont accessibles. Ceci évite en principe à l'utilisateur de la classe de se soucier de son fonctionnement et de faire des erreurs en changeant directement la valeur de certaines donnée

Exemple se classe :



L'architecture de la déclaration de classe dans Visual C++ donne :

```
class Point
        {   int x ;
            int y ;
            public :
    Point (int, int) ; // constructeur de la classe Point
            .....etc.
        }
```

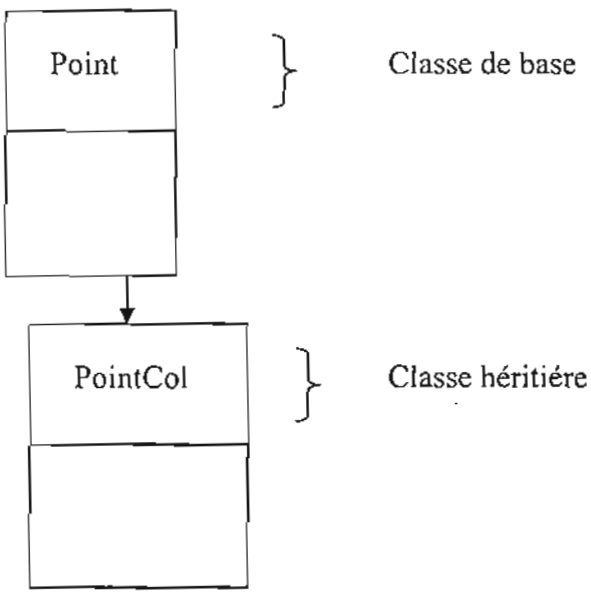
Le constructeur permet de déclarer l'objet et de l'initialiser.

Le destructeur lui permet de détruire l'objet créé.

I. 2 La notion d'héritage :

L'héritage permet de définir une nouvelle classe (fille), qui héritera donc des caractéristiques de la classe de base .Le terme caractéristique inclut en fait l'ensemble de la définition de cette classe mère.

D'un point de vue pratique, il faut savoir qu'il n'est pas nécessaire pour la classe fille de connaître l'implémentation de la base : sa définition suffit. De plus on peut hériter plusieurs fois de la même classe, et une classe fille pourra également servir de base pour une autre classe. Il est alors possible de décrire un véritable a « arbre d'héritage ».



Exemple de classe fille :

```
class PointCol : public Point     // notre classe hérite de la classe
Point
{
   Unsigned char byRed ;
   Unsigned char byGreen;
   Unsigned char byBlue;
public :
    void Colore(unsigned char R, unsigned char G,
   unsigned char B)
}
```

NB: en déclarant un objet de type PointCol, il est possible d'accéder aux membres publics de PointCol, mais également de Point.
Dans les règles de l'art, on sépare la définition de la classe de l'implémentation (un fichier « .h » et un fichier « .cpp »).

I. 3 Notion d'encapsulation des données :

Le concept d'encapsulation est un mécanisme consistant à rassembler les données et les méthodes au sein d'une structure en cachant l'implémentation de l'objet, c'est-à-dire en empêchant l'accès aux données par un autre moyen que les services proposés.
L'encapsulation permet donc de garantir l'intégrité des données contenues dans l'objet.
L'utilisateur d'une classe n'a pas forcément à savoir de quelle façon sont structurées les données dans l'objet, cela signifie qu'un utilisateur n'a pas à connaître l'implémentation. Ainsi, en interdisant l'utilisateur de modifier directement les attributs, et en l'obligeant à utiliser les fonctions définies pour les modifier (appelées interfaces), on est capable de s'assurer de l'intégrité des données (on pourra par exemple s'assurer que le type des données fournies est conforme à nos attentes, ou encore que les données se trouvent bien dans l'intervalle attendu).
L'encapsulation permet de définir des niveaux de visibilité des éléments de la classe. Ces niveaux de visibilité définissent les droits d'accès aux données selon que l'on y accède par une méthode de la classe elle-même, d'une classe héritière, ou bien d'une classe quelconque.
Il existe trois niveaux de visibilité:

- public: les fonctions de toutes les classes peuvent accéder aux données ou aux méthodes d'une classe définie avec le niveau de visibilité public. Il s'agit du plus bas niveau de protection des données
- protégée: l'accès aux données est réservé aux fonctions des classes héritières, c'est-à-dire par les fonctions membres de la classe ainsi que des classes dérivées
- privée: l'accès aux données est limité aux méthodes de la classe elle-même. Il s'agit du niveau de protection des données le plus élevé

II Présentation du problème

II. 1Cahier de charge :

L'interface graphique pourvu du CCEF devra permettre à un utilisateur de faire un traitement particulier sur une structure qu'il aura définie.

Les actions que l'utilisateur aura à effectuer sont :

- Définir la géométrie de la structure (en 2D);
- Préciser la nature des appuis ;
- Etablir un (des) chargement(s) ;
- Recueillir les résultats sur une feuille de propriété ;

L'ensemble de ces données utilisateurs sera traduit en données MEF par les différents modules du solveur.

II. 2 Différents modules d'un solveur éléments finis et leurs taches principales

Un solveur éléments finis est essentiellement constitué de trois composantes :

Le mailleur : (coordonnées et numérotation globale des nœuds de chaque élément)

Ceci nécessite une sous-routine d'interface qui permet de lire le fichier généré à partir d'une boite de dialogue et de fournir les informations sous forme de tableaux : un tableau pour la connectivité des éléments et un autre pour les coordonnées des éléments.

Ceci aura pour but de calculer les matrices L_i de localisation globale de chaque élément mais aussi la matrice jacobienne J de transformation de l'élément de référence à l'élément réel.

Le code éléments finis :

Les différentes étapes de construction du code de calcul aux éléments finis (CCEF) sont :

- Calculs élémentaires : matrices de rigidité, matrice élémentaire associée à une condition aux limites, vecteurs élémentaires de charge, etc.
- Assemblage de la matrice de rigidité et du vecteur de charges globales
- Prise en compte des conditions aux limites de Neumann (CL)

- Prise en compte des conditions aux limites de Dirichlet (CL)
- Résolution du système linéaire

Le post-traitement : visualisation du maillage et de la solution approchée et calcul éventuel des contraintes et déformations.

III Déroulement d'un calcul statique linéaire :

Un programme de calcul statique (PCS) linéaire est conçu pour déterminer le champ de déplacement, les réactions aux appuis, les efforts internes aux noeuds et le champ de contraintes qui existe dans une structure mécanique soumise à divers chargements statiques
Plusieurs hypothèses sont faites implicitement :

- comportement élastique linéaire des matériaux,
- petites déformations,
- petites rotations,

Il existe une relation d'équivalence dans le cas de l'approche cinématique entre la stationnarité de l'énergie potentielle totale et la résolution du système :

$$[K]\{d\} = \{R\} \qquad \text{(II.1)}$$

Où :

$[K]$ est la matrice de raideur de la structure discrétisée,

$\{d\}$ le vecteur des degrés de liberté indépendants de la structure,

$\{R\}$ le vecteur des charges appliquées sur la structure.

Les charges peuvent être de deux types :

- charges ponctuelles (P) ;
- charges reparties (Q) ;

Les charges ponctuelles ou charges nodales sont les charges que l'utilisateur introduit explicitement sur certains noeuds du maillage : ce sont des forces et des moments ponctuels, qui ne sont pas interprétés par le programme en terme de charges énergétiquement

équivalentes en utilisant la fonction de forme de l'élément chargé car la charge est directement affectée au degré de liberté concernée.

Les charges reparties sont des charges de surface ou des charges de volume, dont la répartition nodale est fastidieuse: pression, pression hydrostatique, force centrifuge, accélération dans une direction quelconque, charges d'origine thermique.

Le programme les transforme en charges ponctuelles en utilisant les fonctions de forme des éléments sollicités:elles sont localement fausses partout, mais globalement justes sur chaque élément au sens de l'énergie.

III. 1 Procédure détaillée du calcul :

Un calcul statique linéaire comporte six phases distinctes:

- traduction des données utilisateur en données Eléments finis,
- génération de la matrice de raideur K_e de chaque élément et des charges nodales équivalentes,
- assemblage de la matrice de raideur structurale et du vecteur de charge,
- traitement des fixations et inversion de la matrice de raideur structurale,
- détermination des déplacements : $\{d\} = [K]^{-1}\{R\}$
- calcul des contraintes,

Certaines opérations, comme le traitement des fixations, peuvent se faire directement pendant la phase d'assemblage et ne sont pas séparées.

III. 2 Traduction des données utilisateur en données éléments finis :

Un grand nombre de vérifications peuvent être faites graphiquement dans le pré-processeur du module de calcul:topologie, conditions aux limites, caractéristiques physiques et matérielles.

C'est pendant cette première phase du calcul que l'on réorganise soit la numérotation des noeuds, soit la numérotation des éléments, que l'on attribue et que l'on numérote les degrés de liberté aux noeuds, que l'on repère les degrés de libertés fixés car ils subiront pendant le calcul un traitement particulier.

Les matrices sont très souvent de grande taille($(n \times n)$(où n est égal au produit du nombre de nœuds de la structure et du nombre de degrés de liberté sur chaque noeud), il est donc important qu'elles soient les plus diagonales possibles; il en résultera un gain d'espace disque

pendant la phase de résolution, de temps de calcul, de place requis pour le stockage de l'information, d'entrée/sortie(I/O)...il faut donc numéroter correctement soit les noeuds soit les éléments selon la méthode utilisée par le module de calcul. Il existe plusieurs algorithmes d'optimisation topologique des matrices dont la description n'entre pas dans le cadre de notre projet. Les méthodes d'optimisation ne traite qu'une partie de l'arborescence et rien ne prouve que l'optimum trouvé soit absolu.

La forme de la structure, les chargements, les fixations influent sur le résultat de la réorganisation.

Il pourrait être intéressant que le programme essaie plusieurs méthodes pour trouver la mieux adaptée à une application particulière.

III. 3 Génération des matrices élémentaires :

Le programme calcule la matrice de raideur de chaque élément, dans un système d'axes conventionnels, attachée à l'élément et fonction de l'ordre des noeuds utilisés pour sa définition. Ces axes sont dits axes propres.

Dans le cas des membranes ou des coques par exemple ,une solution consiste à prendre pour x du premier au second noeud,y lui étant orthogonal dans le demi-plan contenant le troisième nœud, z est orthogonal à x et y et forme avec eux un repère direct.

Les déplacements à l'intérieur de l'élément $\mathbf{u} = \{u, v, w\}$ sont interpolés par les déplacements aux nœuds $\mathbf{d}$:

$$\mathbf{u} = \mathbf{N}\mathbf{d} \qquad (II.2)$$

où $\mathbf{N}$ est la matrice des fonctions d'interpolation.

Le champ de déformation est exprimé en terme de déplacements aux nœuds par la relation

$$\varepsilon = \mathbf{B}\mathbf{d} \qquad (II.3)$$

où B dérive de $\mathbf{N}$ par la relation $\mathbf{B} = \mathbf{L}\mathbf{N}$, avec $\mathbf{L}$ un opérateur de dérivation.

Dans le cas d'un élément quadrilatéral de plaque à quatre nœuds défini en 2D :

$$[U(x,y)] = \begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} \qquad (II.4)$$

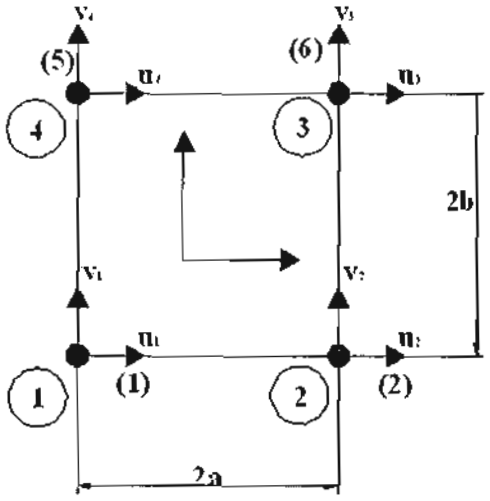


Fig.II.1 : élément de référence rectangulaire

La matrice des fonctions d'interpolation aux nœuds de l'élément est donnée en fonctions des coordonnées paramétriques (ξ, η) par les relations suivantes :

$$[N_e]=\begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \quad \text{(II.5)}$$

avec

$$N_1 := \frac{1}{4}(\eta - 1)(\xi - 1) \qquad N_2 := -\frac{1}{4}(\eta - 1)(1 + \xi)$$

$$N_3 := \frac{1}{4}(1 + \eta)(1 + \xi) \qquad N_4 := -\frac{1}{4}(1 + \eta)(\xi - 1)$$

La matrice de dérivation $[L]$ est donnée par la loi de comportement :

$$[L]=\begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \quad \text{(II.6)}$$

La matrice [B] appelée matrice déformations – déplacements est donnée par :

$$[B]=[L][N] \quad \text{(II.7)}$$

La matrice de rigidité de l'élément est obtenue en faisant l'intégrale :

$$k = \int_{\Omega} B^T H B \, d\Omega = \iint B^T H B \, t dx dy = \int_{-1}^{1}\int_{-1}^{1} B^T H B \, t J d\xi d\eta \quad \text{(II.8)}$$

Où t est l'épaisseur de notre élément et j le déterminant de sa matrice jacobienne.
Et la matrice de rigidité est donnée en annexe dans le tableau (XXX)

III. 4 Assemblage :

Le programme effectue l'assemblage, c'est à dire la construction de la matrice de raideur globale de la structure. Cette opération matricielle est basée sur la méthode des déplacements ou méthodes des matrices de raideur. C'est une méthode matricielle indépendante de la notion d'élément fini, qui permet de raccorder plusieurs éléments sur un même noeud par égalité des déplacements en ce noeud.

Cette méthode a été utilisée pour construire la matrice de raideur des premiers éléments finis, mais elle ne sert aujourd'hui que pour l'assemblage.

Du fait de la continuité des déplacements aux noeuds, on montre que la raideur en un noeud N est la somme des raideurs projetées des éléments qui y sont connectés.

Apres assemblage, la taille du problème est égale au nombre de degrés de liberté indépendants de la structure.

L'assemblage permet d'obtenir la matrice globale de la structure $[K]$ et le vecteur chargement $\{R\}$:

à partir des matrices de rigidité et de chargements des n éléments qui constituent la structure :

$$k^i d^i = r^i_e \quad \text{(II.9)}$$

Pour obtenir les éléments K_{ij} de la matrice globale, on élargie d'abord, toutes les matrices de rigidité élémentaire aux dimensions de la matrice globale et on additionne les éléments $K_{G_{ij}}$ des n éléments. Ils sont donnés par :

$$K_{ij} = \sum_{nel=1}^{n} K_{G\ ij}^{nel} \quad \text{(II.10)}$$

Où les $K_{G_{ij}}^{n}$ sont les éléments de la matrice de rigidité de l'élément n ramené à la matrice de rigidité globale.

Pour le vecteur chargement global de la structure, il est construit en initialisant d'abord ces éléments par les efforts directement appliqués aux nœuds, ensuite on fait la somme des efforts aux nœuds des éléments après avoir ramené les vecteurs chargements de chaque élément aux dimensions d vecteurs de chargement global.

Ces éléments sont donnés par :

$$R_{ij} = \sum_{l}^{n} r_{G\ ij}^{nel} + P \quad \text{(II.11)}$$

P représente les charges directement appliquées aux nœuds.

La procédure de réalisation de cet assemblage est montrée sur la figure (Fig. II.2) :

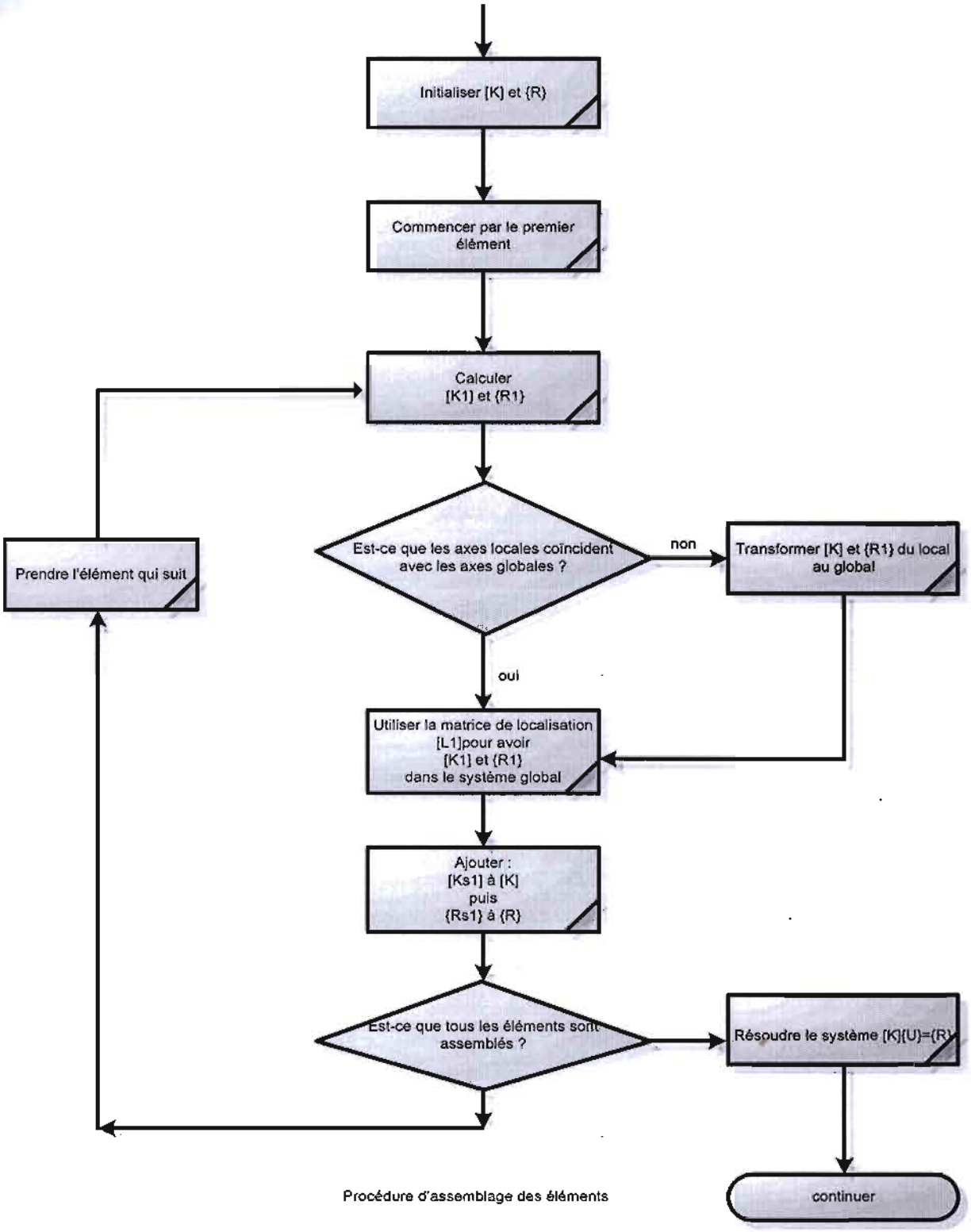


Fig. II.2 : Procédure d'assemblage des éléments de la structure

III. 5 Inversion

Pour calculer le déplacement des noeuds de la structure, quelque soit la méthode utilisée, lorsque la matrice $[K]$ est inversible et bien conditionnée, le programme effectue formellement l'opération : $\{d\} = [K]^{-1}\{R\}$

La matrice $[K]$ est de grande taille et son inversion ne peut pas être réalisée en utilisant les formules classiques d'inversion matricielle (calcul du déterminant et des co-matrices).

On a donc recours à différentes techniques, dont la méthode d'élimination de Gauss.

C'est une méthode qui permet de ramener le système carré initial à un système triangulaire équivalent:

$$[K]\{d\} = \{R\} \Rightarrow [L_1][K]\{d\} = [L_1]\{R\} \Rightarrow [L][K]\{d\} = [L]\{R\} \Rightarrow [T]\{d\} = \{y\} \quad \text{(II.12)}$$

A chaque étape de l'élimination de Gauss, les combinaisons linéaires effectuées sur les lignes et les colonnes modifient la matrice $[K]$ initiale pour la transformer en $[T]$ et le vecteur des charges $\{R\}$ pour le transformer en $\{y\}$: la matrice $[L]$ n'est jamais explicitement formée.

Cette méthode n'est utilisable que si la matrice de raideur de la structure est inversible, c'est à dire si $[K]^{-1}$ existe (ce qui se traduit mathématiquement par un déterminant non nul) et si, en cours d'élimination, un des pivot ne devient pas <<trop petit>>.

Pour de très gros problèmes, dont la taille dépasse 200000 degrés de liberté, les méthodes itératives semblent plus performantes.

III. 6 Traitement des fixations

Dans un calcul strictement linéaire, la fixation d'un degré de liberté est nécessairement bilatérale car il n' y a pas d'ajustement de la condition aux limites selon le signe de la réaction: il n'y a qu'un calcul pour la détermination du champ de déplacement et des réactions.

Le traitement particulier des degrés de liberté fixés à une valeur nulle ou non nulle est effectué pendant cette phase de calcul. On commence par partitionner le vecteur des déplacements et la matrice de raideur associée. On sépare les degrés de liberté fixés à zéro des autres. La relation matricielle s'écrit.

$$\begin{bmatrix} F_l \\ F_f \end{bmatrix} = \begin{bmatrix} K_{ll} & K_{lf} \\ K_{fl} & K_{ff} \end{bmatrix} \begin{bmatrix} d_l \\ 0 \end{bmatrix} \qquad \text{(II.13)}$$

On peut calculer q_l à partir de la première ligne de ce système matriciel.

$$d_l = K_{ll}^{-1} F_l \qquad \text{(II.14)}$$

Seule la matrice K_{ll} est à inverser. Mais si les conditions aux limites changent, le découpage initial du vecteur des degrés de liberté (d_l, d_f) n'est plus valable ni celui de la sous-matrice K_{ll} : il faut la réévaluer et la ré-inverser.

Du fait de la technique de résolution, on ne sait pas traiter de manière classique plusieurs jeux de conditions aux limites en un seul passage alors qu'il est possible d'appliquer plusieurs charges indépendantes.

F_f est un résultat de calcul et n'est pas une donnée :

$$K_{ll} = K_{fl} d_l = K_{fl} K_{ll} F_l \qquad \text{(II.15)}$$

C'est le vecteur des forces qu'il faut appliquer sur les degrés de liberté d_f pour qu'ils ne se déplacent pas : F_f est le vecteur des réactions.

Dans le cas de déplacements imposés, le traitement est légèrement différent car le système s'écrit :

$$\begin{bmatrix} F_l \\ F_i \end{bmatrix} = \begin{bmatrix} K_{ll} & K_{li} \\ K_{il} & K_{ii} \end{bmatrix} \begin{bmatrix} d_l \\ d_i \end{bmatrix} \Rightarrow F_l = K_{ll} d_l + K_{li} d_i \Rightarrow d_l = K_{ll}^{-1} (F_l - K_{li} d_l)$$

(II.16)

Les charges Fi sont les efforts qu'il faut appliquer sur les degrés de liberté d_i pour qu'ils se déplacent de la quantité voulue:

$$F_i = K_{il} d_l + K_{ii} d_i \qquad \text{(II.17)}$$

Il est tout à fait possible d'avoir des degrés de liberté fixés et d'autres à déplacements imposés dans un même calcul :il suffit de partitionner de manière adéquate les matrices et vecteurs. Les matrices étant très souvent d'une taille telle qu'elles ne tiennent pas entièrement en mémoire centrale, il est nécessaire de recourir à un découpage de ces matrices et de les stocker sur disque par morceaux.

Il existe deux types de méthodes pour inverser une matrice :

La méthode frontale par blocs est une méthode d'élimination de Gauss qui tient compte de la topologie de la matrice de raideur et pour laquelle il est important d'optimiser la numérotation des éléments, ce qui conduit à la notion de largeur de front.

la méthode de la <<ligne de ciel>>(ou skyline) est aussi une méthode d'élimination de Gauss où on tient compte du caractère creux de la matrice de raideur en ne stockant ,pour chaque colonne ,que les termes compris entre la diagonale et le dernier terme non nul de la colonne et pour laquelle il est important d'optimiser la numérotation des noeuds ,ce qui conduit à la notion de largeur de bande.

III. 7 Détermination des déplacements

Après partitionnement du problème, le système à résoudre est

$$F_I = K_{II} d_I \qquad \text{(II.18)}$$

que l'on triangularise par la méthode d'élimination de Gauss.

Une fois le système sous forme triangulaire, sa résolution ne pose aucun problème.

La dernière ligne permet la détermination du déplacement du degré de liberté d_n.

$$t_{nn} d_n = y_n \quad \Rightarrow \quad d_n = \frac{1}{t_{nn}} y_n \qquad \text{(II.19)}$$

La ligne du dessus, connaissant d_n, permet la détermination de d_{n-1} :

$$t_{n-1n-1} d_{n-1} + t_{n-1} d_n = y_{n-1} \Rightarrow d_{n-1} = \frac{1}{t_{n-1n-1}} (y_{n-1} - t_{n-1n} d_n) \qquad \text{(II.20)}$$

De proche en proche, par substitution inverse, on calcule les inconnues d_i par la relation de récurrence :

$$d_i = \frac{1}{t_{ii}}(y_i - \sum_{r=i+1}^{n} t_{ir} d_r) \qquad \text{(II.21)}$$

III. 8 Calcul des contraintes

Le calcul des contraintes était initialement effectué sans passer par celui des déformations. En effet, la contrainte σ est reliée aux déplacements discrets par :

$$\sigma = H\varepsilon = HBd = Td \qquad \text{(II.22)}$$

T est la matrice des contraintes qui permet de calculer les contraintes directement à partir des déplacements aux noeuds. En fait le retour aux contraintes dépend du type d'élément fini utilisé pour le modèle et en pratique, on effectue directement le produit $\sigma = H\varepsilon$ car il coûte moins au niveau informatique.

IV. Les grands concepts de l'interface graphique d'un CCEF

Afin de bien comprendre le pré développement d'un code éléments finis, il faut d'abord illustrer les concepts qui y seront implantés ainsi que les relations entre ces concepts. Étant donné qu'il y a plusieurs aspects importants dans la résolution d'un problème par éléments finis, il faut en comprendre un minimum afin de pouvoir utiliser correctement les outils développés dans les programmes.

Le CCEF (code de calcul par éléments finis) devrait toucher pratiquement à tous les aspects d'un problème d'éléments finis et doit posséder des classes et des fonctions permettant le traitement de chacun de ces aspects. Le CCEF est pré développé dans un langage orienté objets et utilise les avantages de cette approche.

Vous trouverez dans les pages suivantes une description la plus visuelle possible des principales classes présentes dans le CCEF ainsi que de leurs rôles et de leurs interactions.

Les paragraphes qui suivent mettent en détails les différentes parties du CCEF : « Les données d'entrées d'un problème » et « Formulation variationnelle et résolution d'un problème ».

IV.1 Les données d'entrées d'un problème

Cette partie du document fait une description des concepts reliés aux données d'entrée pour la résolution d'un problème. Nous mettrons en évidence les liens entre les différents concepts utilisés.

IV. 2 Vue d'ensemble de la résolution d'un problème

Avant de commencer à expliquer tous les concepts qui existent dans le code, voici une description de la vue globale de ce que l'on peut avoir du code :

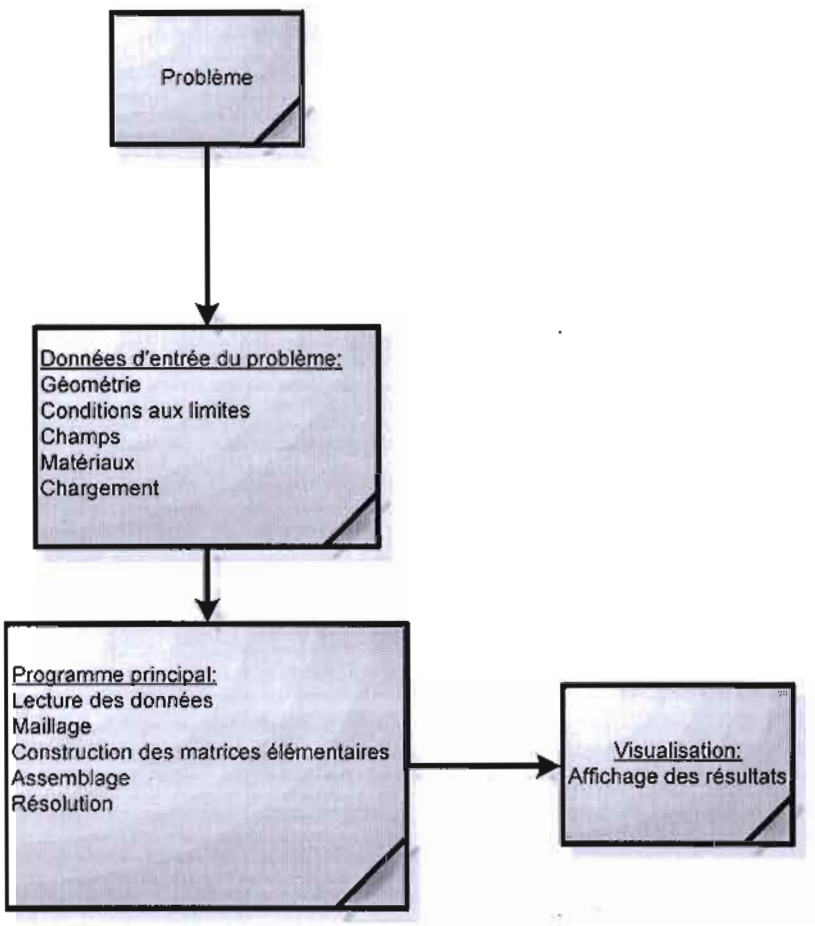


Fig. II.3: Vue globale de la résolution d'un problème

- Tout commence lorsque vous avez un problème à résoudre. Il vous faut d'abord formuler ce problème, poser les équations qui entrent en jeu, décrire les paramètres du

problèmes (type de matériau, constantes du problème, etc.) et les conditions aux limites.

- Ensuite, si vous choisissez de résoudre votre problème à l'aide de la méthode des éléments finis, il faut faudra discrétiser le domaine à simuler. À la fin de cette étape, vous aurez au minimum un maillage d'éléments qui ne contient que des coordonnées et des connectivités des éléments.

- Il vous faut maintenant lire le maillage que vous avez créé à l'étape précédente et « recréer » la géométrie sur ce maillage.

- C'est sur cette géométrie que nous travaillerons pour la définition des conditions aux limites et des matériaux. Sur la géométrie, les endroits qui nous intéressent seront identifiés comme des « entités géométriques » (nœuds, arrêtes, faces etc.).

- À l'aide du CCEF, on écrit (exporte) les données nécessaires à la résolution : la géométrie (.geom), le maillage (.mail) , les entités géométriques (.enti). Ensuite, on va écrire un fichier de conditions aux limites (.CL) ainsi qu'un fichier de champs et de matériaux (.champs). Ces fichiers sont typiquement ce que tout programme principal demande pour une résolution de problème.

- Ensuite, on lance la sous routine de calculer. Celle-ci fera la lecture des fichiers d'entrée, fera les assemblages nécessaires ainsi que la résolution du système obtenu. Une fois ce travail complété, il va écrire (exporter) les résultats pour permettre la visualisation.

- Pour la visualisation, on peut utiliser la fenêtre vue en mode texte (Edit_View) ou tout autre visualisateur supporté par le code tel que le bloc notes (WORDPAD).

- Une fois le problème résolu, on peut maintenant valider les résultats avec la pratique ou avec la théorie.

IV. 3 La géométrie

La géométrie est un aspect fondamental dans un programme de calcul structural. La géométrie est la représentation de l'objet à discrétiser (Fig.II.4). La géométrie proprement dite ne contient pas de maillage. C'est une « définition » de l'objet que l'on veut représenter. Par exemple, on va parler de la géométrie d'une dalle, qui consistera à représenter tous les détails de cette dernière par des points, des courbes et des surfaces suffisamment riches pour bien la représenter.

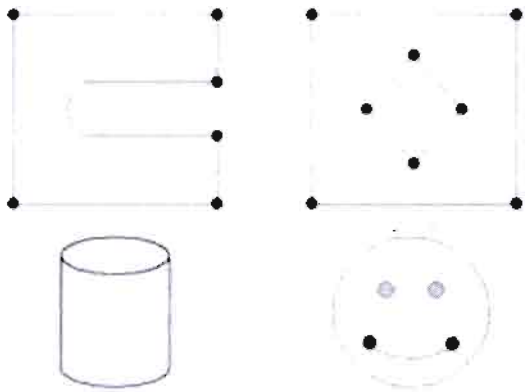


Fig. II.4 : exemple de géométrie

La représentation graphique de la structure dans un PCS peut être effectuée en utilisant la classe<< CGéométrie>> comme classe de base à partir de laquelle ,dérivent les classes <<CRect>>,<< Cpt>>, <<CCercle>< et <<CSegment>> pour le dessin des formes régulières comme les points ,les segments de droites ,les rectangles et les cercles.

Un accent particulier doit être mis sur la saisie graphique avec des classes hiérarchisées comme la montre la figure (Fig.xxx).

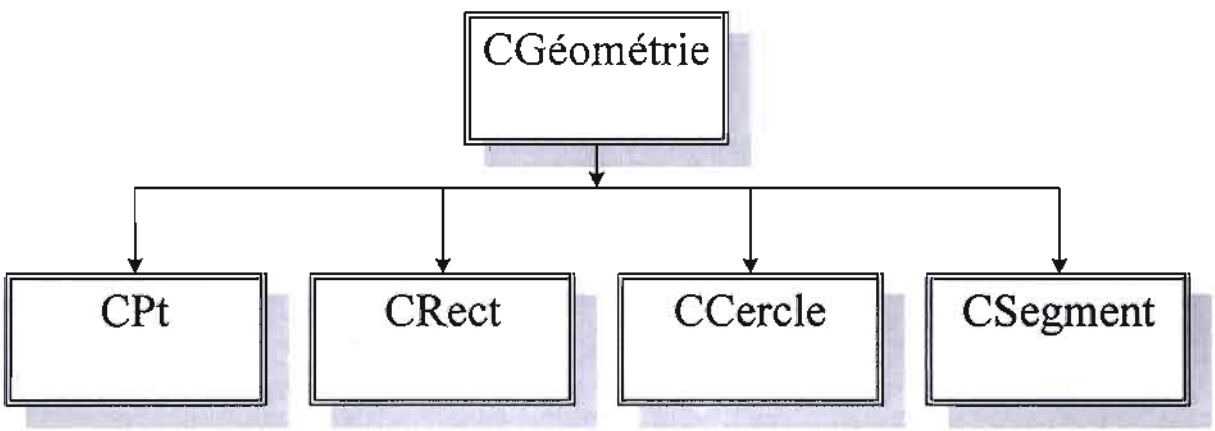


Fig. II.5 Hiérarchie des classes de dessin de la structure

et le dessin se faisant au moyen d'une commande polymorphe ,les objets ainsi dessinés sont listés dans un vecteur (m_geometrie) déclaré à cet effet (CTypedPtrList <CObList ,CGeometrie*> m_geometrie).
Chacune des classes de dessins permet de représenter graphiquement des objets définis comme entités géométriques.

Étant donné une géométrie, on peut avoir plusieurs maillages qui s'y rattachent (Fig.II.5). Une géométrie est donc a priori indépendante des maillages. Un maillage se veut donc une discrétisation particulière d'une géométrie donnée. De plus, le CCEF doit inclure des fonctionnalités pour gérer les maillages qui portent sur une même géométrie.

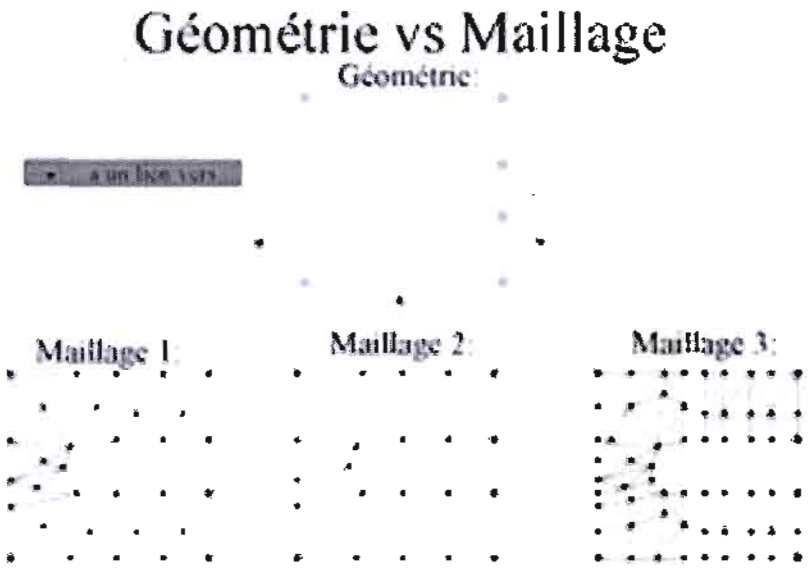


Fig.II.6 : une géométrie avec des maillages différents

Dans le CCEF, chacun de ces points, courbes, surfaces et volumes est ce que l'on appelle un lieu géométrique (la classe « LieuGeometrique »).

IV. 4 Entité géométrique

La notion d'entité géométrique est simplement un regroupement de lieux géométriques (Fig.II.8). C'est une notion qui représente un groupe de lieux géométriques et qui est surtout utilisée pour imposer les conditions aux limites et faire des associations de matériaux. Une entité portera le nom que l'usager aura choisi et qui pourra représenter le nom du morceau de la pièce qu'il représente (Ex : « Entrée » pour l'entrée d'un tuyau).

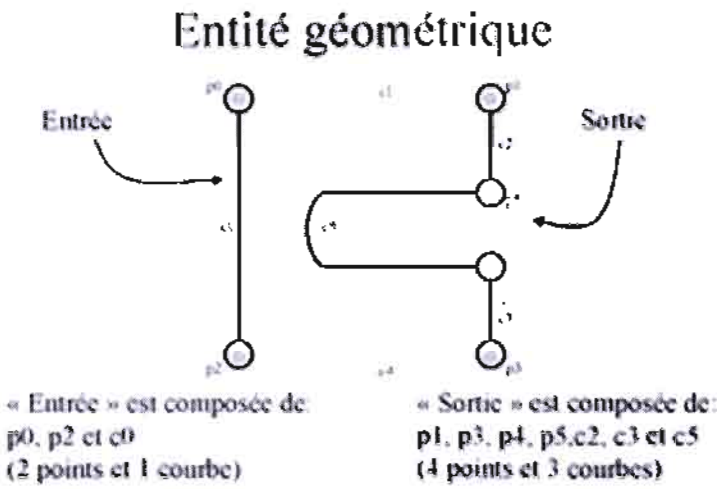


Fig. II.7: entité géométrique

Les entités géométriques sont en lien direct avec la géométrie sur laquelle elles sont définies. Une géométrie peut avoir plusieurs entités géométriques définies sur elle-même. Tous les maillages reliés à cette géométrie peuvent utiliser automatiquement les entités définies sur cette dernière, c'est-à-dire que, si l'on définit l'entité géométrique « Entrée » qui est constituée de 2 points et de 1 courbe, lorsque l'on décide d'appliquer une condition aux limites sur cette entité, les bons nœuds de calcul associés à cette entité par l'intermédiaire du maillage et du champ sont automatiquement sélectionnés.

L'avantage de travailler avec une géométrie et des entités géométriques est que, si vous préparez les données pour faire une simulation, ces données sont conservées sans être en lien direct avec un maillage. Si vous changez de maillage, vous pouvez donc réutiliser les données de votre problème (conditions aux limites, matériaux, changements de repère), ce qui vous permet de tester rapidement plusieurs maillages pour les mêmes conditions de simulation.

IV. 5 Relation Géométrie / Matériaux

Dans un CCEF, il est aussi possible de traiter des problèmes avec différents matériaux. Comme les conditions aux limites, les matériaux sont associés à des entités géométriques (Fig.II.9). Donc on pourra réutiliser la même distribution de matériaux pour plusieurs maillages, pour autant qu'ils portent tous sur la même géométrie.

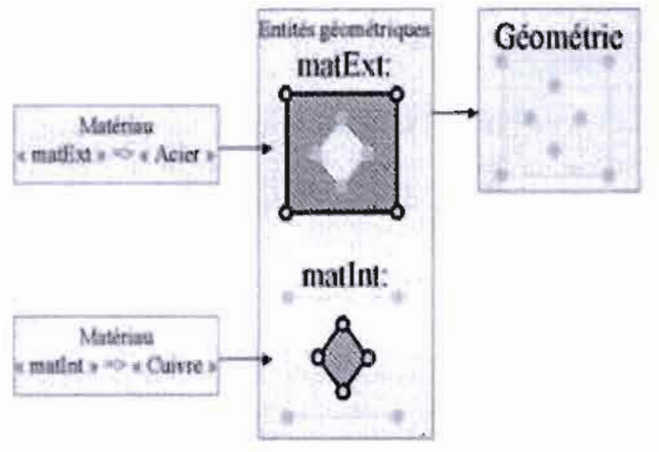


Fig.II.8 : relation entre la géométrie, les entités et les matériaux

IV. 6 Relations Entités géométriques / Conditions aux limites

Maintenant que vous avez vu l'avantage de travailler avec une géométrie et des entités géométriques plutôt que directement avec un maillage, voyons ce que vous pouvez faire avec des entités géométriques.

La première utilisation d'une entité géométrique est l'imposition de conditions aux limites. Pour imposer des conditions aux limites à un problème, il faut d'abord créer les entités géométriques afin de regrouper les morceaux de géométrie qui porteront ces conditions aux limites. En ayant choisi des noms représentatifs (ex. : « Entrée », « Sortie », « Haut », « Bas », etc.) pour des entités géométriques, le lien avec les conditions aux limites devient plus près du langage écrit.

Vous remarquerez que, jusqu'ici, on n'a pas parlé d'interpolation du champ auquel est reliée la condition aux limites. En fait, les conditions limites sont a priori indépendantes du type d'interpolation choisi, donc indépendantes d'un champ en particulier. C'est seulement lorsque l'on affectera des conditions aux limites pour un problème précis que celles-ci seront imposées aux bons nœuds de calcul.

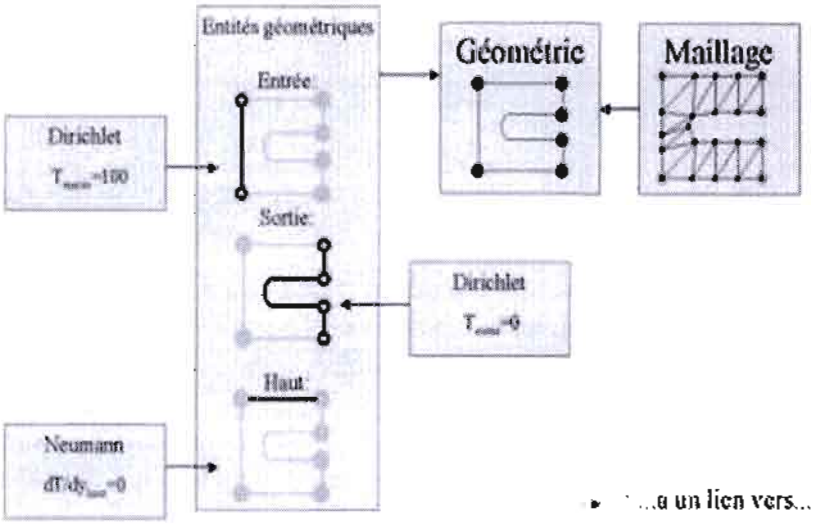


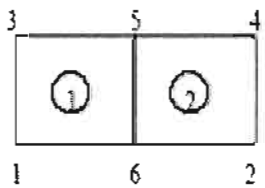
Fig.II.9 : relation entre la géométrie, les entités et les conditions aux limites

IV. 7 Le maillage

Il est constitué de plusieurs listes ou tables. Il contient une liste de noeuds, de coordonnées et d'éléments avec leur connectivité respective.

Chaque élément est défini par la donnée de **nnode** numéros de noeud. On appellera table de connectivité (ou de topologie) d'un élément la liste de ces numéros, notée **lnods** et de dimension **nnode** × **nelem**. Par convention, cette liste est donnée en parcourant les noeuds de l'élément dans le sens trigonométrique. Les coordonnées des noeuds sont rangées dans un tableau **coord** de dimension **ndime** × **npoin**, où **ndime** est la dimension de l'espace (1, 2 ou 3) et **npoin** le nombre de noeuds dans le maillage.

Exemple :



ndime = 2 *nnode* = 4
nelem = 2 *npoin* = 6

Table de *connectivité* :

élément 1 : lnods(*,1) = (1 6 5 3)
élément 2 : lnods(*,2) = (4 5 6 2)

Table des coordonnées

coord(*,1) = (0,0)
coord(*,2) = (2,0)
coord(*,3) = (0,1)
coord(*,4) = (2,1)
coord(*,5) = (1,1)
coord(*,6) = (1,0)

C'est la même chose pour la liste des éléments. On y retrouve les 6 types d'éléments décrits ci-dessous.

Puisque toutes les classes d'éléments héritent du parent « Element », on peut les traiter ainsi. Dans la liste d'éléments, on retrouve les éléments ordonnés par type. L'ordre est le suivant :

- Element1D
- ElemTriangle
- ElemQuad
- ElemTetra
- ElemHexa
- ElemPrismeTri

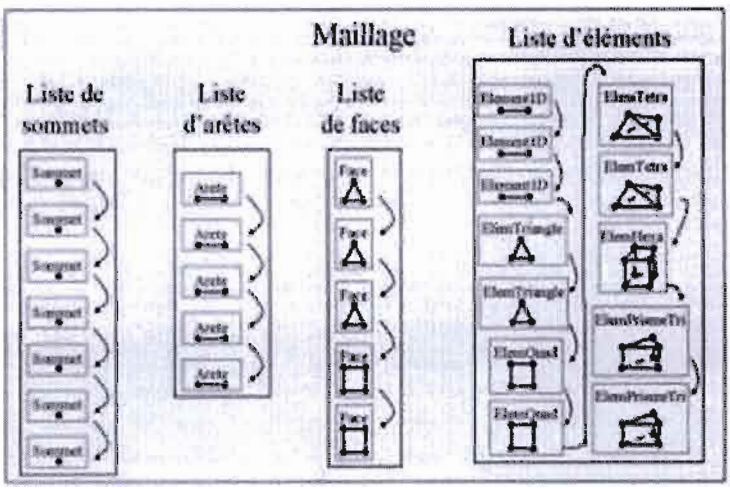


Fig. II.10 : Le contenu de la classe «Maillage»

IV. 8 Hiérarchie d'éléments

IV.8.1 La notion d'élément :

Un élément n'est rien d'autre qu'une forme géométrique (topologie), une connectivité entre des noeuds, des arêtes ou des faces. Il constitue l'élément de base d'un maillage pour une géométrie donnée.

Les informations pouvant être recueillies sur un élément d'une structure sont résumés sur la figure suivante (Fig.xxx):

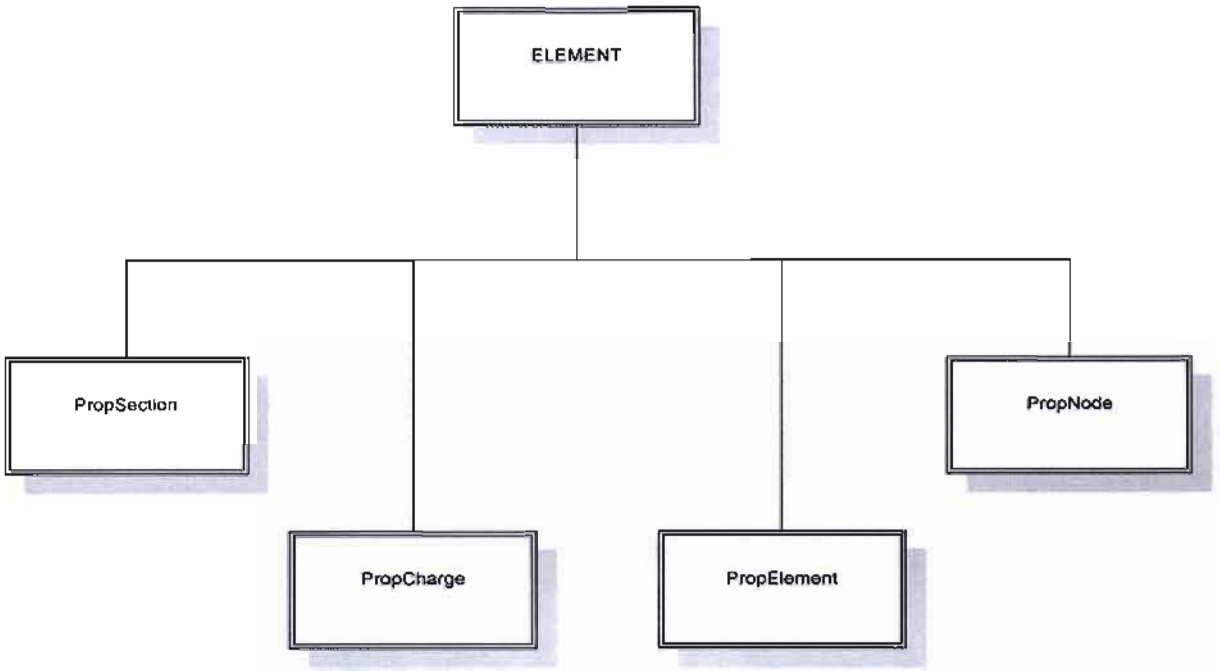


Fig.II.11 : Routines d'informations sur un élément

Chacun des tableaux définis ci haut est une structure de données informatiques dans le programme de construction du module de calcul. Ces différentes structures permettant de définir les caractéristiques d'un élément et les charges qui lui sont appliquées.

La structure <<PropSection>> :

Cette structure de données sert à définir la nature de la section d'un élément de la structure

- Identification

```
CString Nom;                // Nom de la section
CString Type;               // Normées ou personnalisée
CString BaseProfile;        // Base ou type de profilé
```

- Paramètres

```
double ParamGeomtrique[15]; // Paramètres géométriques
double ParamMecanique[10];  // Paramètres mécaniques
```

- Position

```
double Alpha;               // angle d'inclinaison
int   PosNode;              // numéro du noeud définissant l'inclinaison
```

```
double Ey;                  // Excentricité suivant y
double Ez;                  // Excentricité suivant z
double CGy;                 // Position y du centre de gravité
double CGz;                 // Position z du centre de gravité
```

- Ordre

```
int   iSection;             // numéro d'ordre de la section
```

La structure <<PropCharge>>

La structure de données PropCharge permet de définir les caractéristiques du chargement appliqué sur la structure.

- Identification

```
CString Nature;             // Nature de la charge : Exploitation ou
                            //                       Permanente ...
CString Type;               // type de chargement : ponctuel, partiel ou
                                                    Uniforme.
CString NomLoadCase;        // Nom de la Charge
```

- Repère

```
CString Repere;
```

- Position

```
CString Position;
```

- Paramètres

```
UINT  NumCasCharge;         // Numéro du cas de chargement
double ParamCharge[100];    // Paramètres géométriques
```

- Ordre

```
int   iCharge;              // numéro d'ordre de la charge
```

La structure <<PropElement>>

La structure PropElement définit les caractéristiques d'un élément de la structure.

```
int     ElmtCode;           // Code de l'élément
int     ElmtType;           // Type de l'élément
int     ElementOrderNumber; // Numéro d'ordre de l'élément
int     ElmtAssNodesNum;    // Nombre de noeuds de liaison
```

```
int     ElmtGeoNodesNum;        // Nombre de noeuds géométriques
int     TotElmtNodes;           // Nombre total de noeuds(de
                                // Liaison et géométriques)

int     ElmtNodes[ElmtNodesMax];// Numéro des noeuds de l'élément
int     ElmtFacesNumber;        // Nombre de faces de l'élément
int     ElmtSidesNumber;        // Nombre de côtés de l'élément
int     NumberNodesPSides;      // Nombre de noeuds par côté

CRect  rectElement;             // rectangle containing the truss element
CRect  rectElmt;                // rectangle containing the element

double CoorX[ElmtNodesMax];     // vecteur de coordonnées suivant X de l'élément
double CoorY[ElmtNodesMax];     // vecteur de coordonnées suivant Y de l'élément
double CoorZ[ElmtNodesMax];     // vecteur de coordonnées suivant Z de l'élément

int     SCoorX[ElmtNodesMax];
int     SCoorY[ElmtNodesMax];

PropSection     Section[8];     // la section de l'élément
PropCharge      Charge[30];     // la charge appliquée sur l'élément.
int             ChargeCounter;  // décompte des charges appliquées
```

La structure <<PropNode>>

Cette structure nous permet de définir les caractéristiques d'un nœud d'élément.

// Attributes

```
int             NodeOrderNumber;    // Nombre total de noeuds
UINT            NodeRadius;         // rayon du noeud
CPoint          NodeCenter;         // centre du cercle représentant le
                                    // noeud
CRect           rectNode;           // rectangle contenant le noeud
double          CoorX;              // coordonnée X du noeud
double          CoorY;              // coordonnée Y du noeud
double          CoorZ;              // coordonnée Z du noeud
```

```
PropCharge      Charge[30];        // Charges appliqués au noeud
int             ChargeCounter;
```

IV. 9 Les connectivités de base

Dans le modèle objets du CCEF, les éléments, faces, arêtes et sommets ont des liens entre eux. Pour les arêtes, la connectivité est simple. Elle consiste en un lien (pointeur) vers 2 sommets ou noeud. Il est donc possible de demander à une arête les 2 sommets qu'elle « possède ». Il faut noter que l'arête ne conserve qu'un pointeur vers ses 2 sommets. Ces 2 sommets sont des objets séparés de l'arête, qui sont dans la liste des sommets du maillage. Il est nécessaire que chaque sommet soit défini séparément d'une arête, puisqu'ils peuvent être utilisés ou référés par plusieurs arêtes.

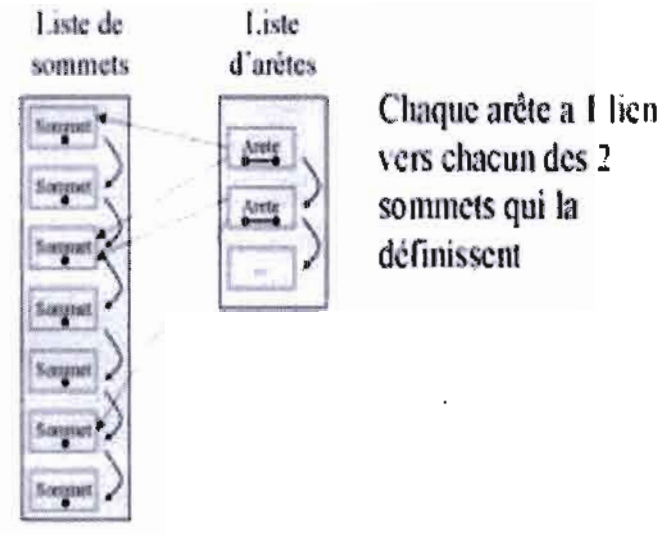


Fig.II.12 : connectivité entre les arêtes et les noeuds

En plus des arêtes, il y a aussi les faces qui, elles, conservent un lien (pointeur) vers les arêtes qui la composent. Pour chaque face, on a donc un nombre de liens qui correspond au nombre d'arêtes qui définissent la face. Il faut noter que les arêtes sont des objets qui existent séparément de la face. La face ne conserve donc qu'un pointeur vers ses arêtes, puisqu'une arête peut être en lien avec plusieurs faces :

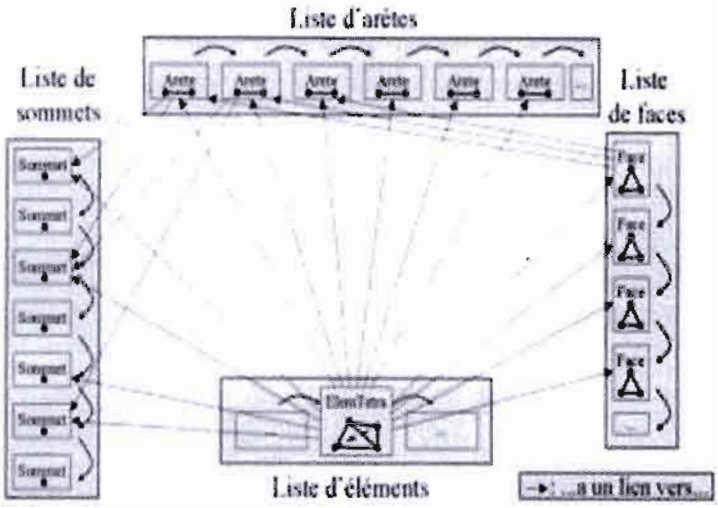


Fig.II.13 : connectivité entre les faces, arêtes et sommets

Finalement, les éléments ont aussi des liens vers les sommets, arêtes et faces qui les composent. Pour un élément unidimensionnel (« Element1D »), il y a des liens seulement vers des sommets. Pour un élément bidimensionnel (« ElemTriangle » et « ElemQuad »), il y a des liens vers des sommets et des arêtes. Pour un élément tridimensionnel (« ElemTetra », « ElemHexa » ou « ElemPrismeTri »), il y a des liens (pointeurs) vers des sommets, arêtes et faces. Donc, pour un tétraèdre, il y aura 4 liens vers des sommets, 6 liens vers des arêtes et 4 liens vers des faces :

En regardant la connectivité d'un élément , vous vous demandez peut-être pourquoi un élément conserve des liens vers ses faces, ses arêtes et ses sommets, alors que s'il conservait uniquement des liens vers ses faces cela serait suffisant. C'est une observation juste. Toutefois, il y aurait un travail supplémentaire à faire pour retrouver les sommets qui composent un élément. Ce travail supplémentaire est dû au fait qu'il faudrait « déduire » quels sont les sommets d'un élément à partir de ses faces à chaque fois, si on ne veut pas conserver de liens vers ces sommets. Pour éviter ce travail, on utilise un peu plus de mémoire.

V Les champs

Qu'est-ce qu'un « champ »? Ce concept très important dans la MEF est au cœur de tous les programmes que vous allez utiliser ou programmer vous-même. Il est donc important de bien comprendre ce concept incontournable.

D'abord, le concept de champ réunit 2 choses :

- les données (ex : degrés de liberté, fonction analytique, propriété de matériau);
- les fonctions (méthodes) que l'on peut utiliser sur ces données.

Tous les champs sans exception sont capables de s'interpoler sur un élément, étant donné une coordonnée de référence.

À ce chapitre, le concept de champ est beaucoup plus qu'un simple « vecteur » de degrés de liberté. Pour les champs avec DDLs, c'est aussi une combinaison, ou liaison entre les degrés de liberté et les fonctions d'interpolation. Pour les champs qui calculent des propriétés physiques, c'est une liaison entre une loi de comportement et les variables dont elle dépend pour se calculer.

Il existe 3 grandes catégories de champs dans un CCEF :

- Les champs de transformation géométrique (ex. : « ChampGeoLin »)
- Les champs éléments finis avec DDL (ex. : « ChampScalLin »)
- Les lois de comportement (ex : « ChampSolideHookeenConstant »)

V. 1 Les champs de transformation géométrique

Maintenant que vous avez vu qu'un élément n'était qu'une forme géométrique, une connectivité, vous vous demandez sans doute comment utiliser ce concept d'élément pour faire, par exemple, des éléments courbes. À la base du code, tous les maillages ont des éléments droits, des arêtes droites et des faces planes. Le concept de transformation

géométrique des éléments est un concept séparé, distinct en soi. Il appartient à l'utilisateur de faire porter ou non une courbure sur l'élément. C'est quelque chose que l'on « ajoute » à un élément géométrique.

La transformation de l'élément réel vers l'élément de référence, pour des éléments droits ou courbes, est un concept qui est disponible dans plusieurs programmes de calcul structural. Ce concept s'appelle champ géométrique (« ChampGeometrique »). Il permet à un élément d'avoir une courbure et permet surtout de calculer la matrice jacobienne de transformation de l'élément réel vers l'élément de référence.

Pour pouvoir avoir des éléments courbes, il faut plusieurs choses :

- le maillage droit;
- la géométrie définie à partir de ce maillage

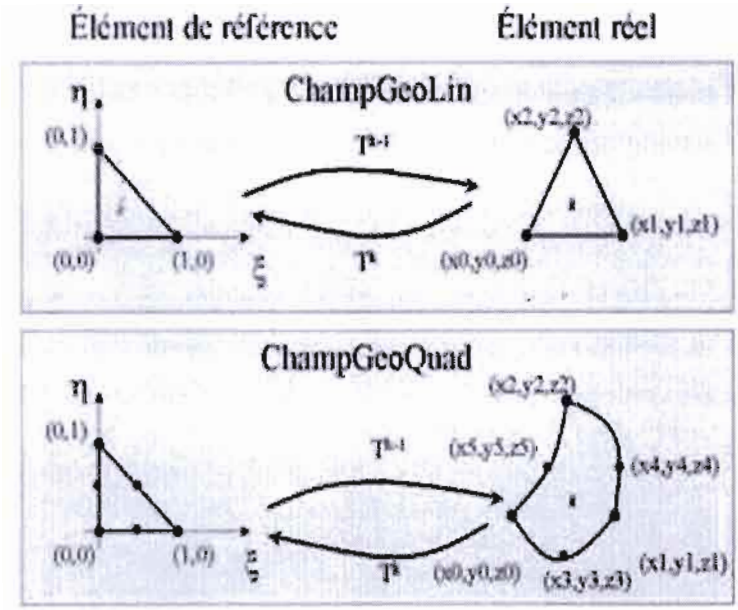


Fig. II.14: les champs de transformation géométrique

Aussi, la fonction la plus importante du champ géométrique est sans doute le calcul des matrices jacobiennes de transformation. Ainsi, étant donné un élément et une coordonnée de référence, le champ géométrique peut donner la matrice jacobienne pour ce point. Cette matrice jacobienne sera utilisée plus tard dans le calcul élémentaire et aussi pour transformer les dérivées premières et secondes des fonctions de base de l'élément de référence vers l'élément réel.

Le champ géométrique possède donc des fonctions d'interpolation qu'il interroge pour pouvoir calculer la matrice jacobienne. On va retrouver le même concept de fonction d'interpolation autant dans le champ géométrique que dans les champs éléments finis. Toutefois, il faut bien comprendre que les fonctions d'interpolation du champ géométrique sont utilisées avec les nœuds géométriques.

V. 2 Les champs éléments finis avec DDL

Maintenant que nous avons une transformation géométrique pour les éléments, il reste 2 concepts importants à ajouter aux éléments géométriques, soit les concepts d'interpolation et de nœuds de calcul.

Bien qu'avec un champ géométrique les éléments aient déjà des fonctions d'interpolation pour la transformation géométrique, celles-ci peuvent être différentes des fonctions d'interpolation utilisées pour la fonction approchée. Elles peuvent être iso-paramétriques, pseudo-isoparamétriques, sub-paramétriques ou super-paramétriques.

Comme la transformation géométrique, les concepts d'interpolation et de noeuds de calcul sont des choses que l'on « ajoute » à un élément géométrique. L'usager peut donc choisir de faire porter une interpolation quadratique avec des degrés de liberté scalaires sur les éléments d'un maillage.

Le nombre de fonctions de base sur un élément définit automatiquement le nombre de nœuds de calcul. C'est donc pour assurer une cohérence que l'on retrouve dans les champs une association intime entre l'interpolation et les nœuds de calcul.

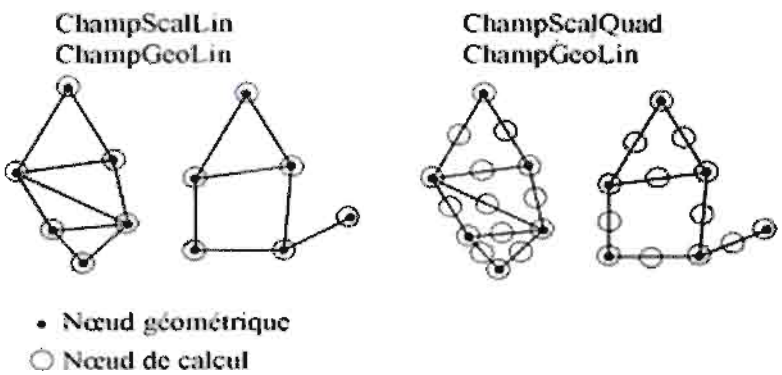


Fig. II.15: exemples de champs d'éléments finis

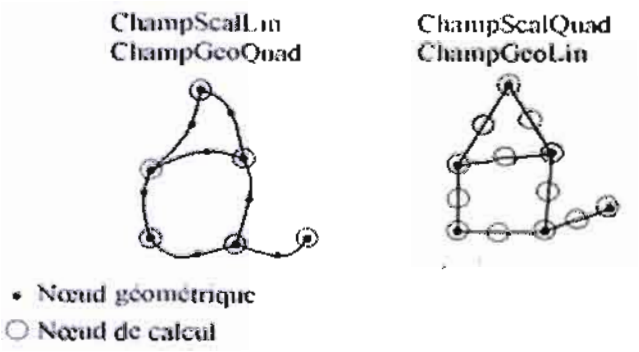


Fig. II.16: Champs combinés

On peut consulter la documentation de la classe pour connaître exactement le type d'interpolation utilisée pour chaque forme d'élément. Par exemple, pour un champ linéaire « ChampScalLin » on peut voir dans la documentation [5] que l'interpolation utilisée est celle donnée par la classe « FonctionsLagrangeLin » qui comporte les bases polynomiales suivantes pour chaque forme d'éléments :

Élément 1D, 2 noeuds: $\langle P \rangle = \langle 1 \quad \xi \rangle$

Élément triangle, 3 noeuds: $\langle P \rangle = \langle 1 \quad \xi \quad \eta \rangle$

Élément quad, 4 noeuds: $\langle P \rangle = \langle 1 \quad \xi \quad \eta \quad \xi\eta \rangle$

Élément tetra, 4 noeuds: $\langle P \rangle = \langle 1 \quad \xi \quad \eta \quad \zeta \rangle$

Élément prisme, 6 noeuds: $\langle P \rangle = \langle 1 \quad \xi \quad \eta \quad \zeta \quad \xi\zeta \quad \eta\zeta \rangle$

Élément hexa, 8 noeuds: $\langle P \rangle = \langle 1 \quad \xi \quad \eta \quad \zeta \quad \xi\eta \quad \eta\zeta \quad \zeta\xi \quad \xi\eta\zeta \rangle$

Plus visuellement, voici une représentation des nœuds de calculs et leurs positions pour les champs dits « linéaires » pour chaque type d'élément :

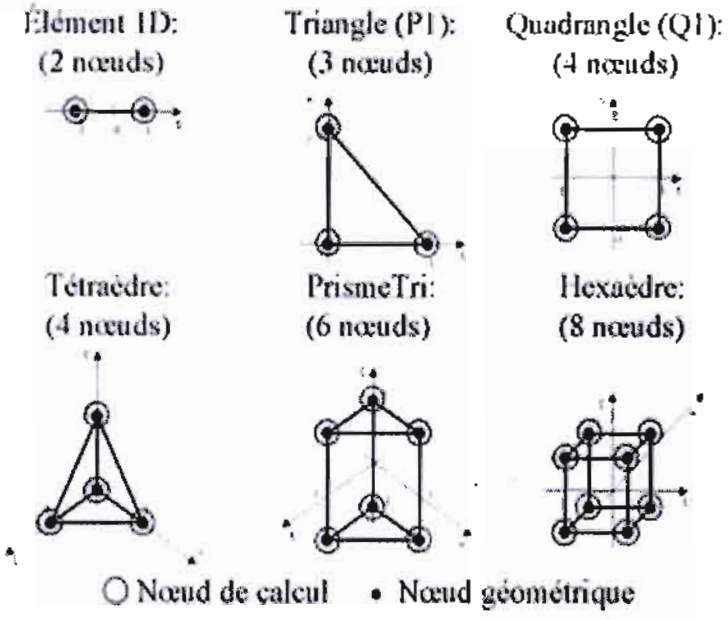


Fig. II.17: Champs linéaires pour tous les types d'élément

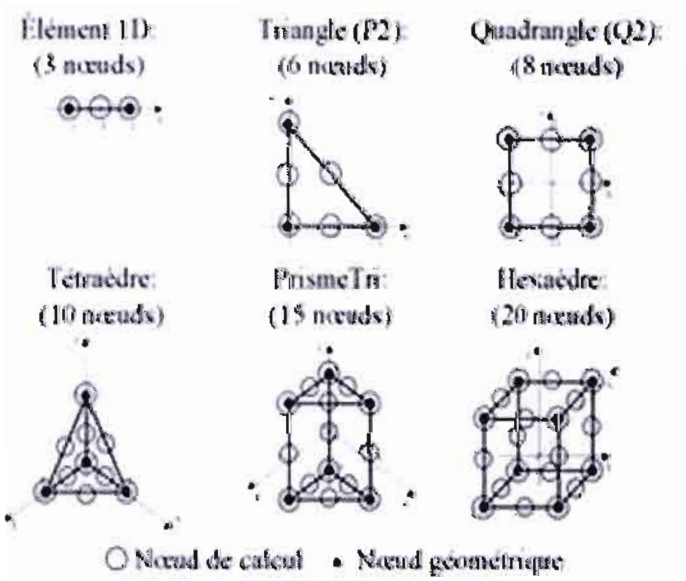


Fig. II.18: champs quadratiques pour tous les types d'éléments

Lorsque nous avons un problème à plusieurs variables, il faut choisir un type d'interpolation pour chaque inconnue. Dans le PCS, l'usager doit être libre de choisir ce qu'il veut comme combinaison car il n'existe pas de combinaisons pré-établies ou imposées. Le code doit permettre justement de changer facilement le type d'interpolation d'une variable de problème pour étudier l'effet sur la solution. Voici quelques exemples de combinaisons :

Dans le cas d'une interpolation mixte (linéaire et quadratique)

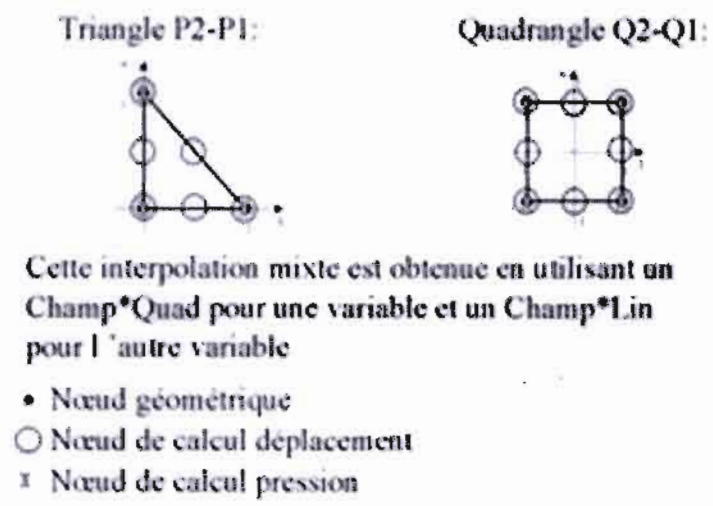


Fig.II.19: interpolation mixte : Quad-Lin

V. 3 Les lois de comportement

Certains champs dans les PCS représentent une loi de comportement de matériau. Ces lois de comportement nous donnent les valeurs de certaines propriétés d'un type de matériau. Ces valeurs pourront par la suite être utilisées dans une formulation variationnelle écrite indépendamment de la loi de comportement.

Voici quelques exemples de lois de comportement :

- Hook
- Neo-Hook
- Mooney Rivlin
- Yeoh

V. 4 Relations Géométrie/Entités Géométriques/Maillage/Champ

Il existe une relation entre les concepts de géométrie, d'entité géométrique, de maillage et de champ. D'abord, tout commence par la notion de géométrie. La géométrie définit la pièce que l'on veut représenter par des points, courbes, surfaces et volumes (« LieuGeometrique »).

Sur cette géométrie, on définit des entités géométriques comme étant des regroupements de lieux géométriques.
Il y a ensuite le maillage. Un maillage n'est qu'une discrétisation d'une géométrie. Comme on l'a déjà vu, il peut exister plusieurs maillages pour une même géométrie. Le maillage qui est rattaché à une géométrie peut être utilisé pour faire un problème complet. Le maillage étant rattaché à une géométrie, cela identifie chacun de ses sommets, arêtes, faces et éléments à un lieu de la géométrie (« LieuGeometrique »).

On ajoute ensuite la notion de transformation géométrique (« ChampGeometrique »). La transformation peut être linéaire ou quadratique. Le champ de transformation géométrique que l'on choisit d'associer à un maillage permettra de transformer les éléments réels vers l'élément de référence.

Enfin, on ajoute les notions de nœud de calcul et d'interpolation (qui sont intimement liées) au maillage et associées au champ de transformation géométrique. La notion d'interpolation permet de calculer les valeurs des fonctions de base, de leurs dérivées premières et secondes, avec et sans transformation sur l'élément réel. Avec des coordonnées de référence, les matrices jacobiennes de transformation de l'élément réel vers l'élément de référence (données par le champ géométrique) et un élément, on peut calculer les fonctions d'interpolation ainsi que leurs dérivées premières et secondes. Une fois les fonctions d'interpolation calculées, on peut s'en servir pour interpoler la valeur d'un champ sur un élément.

Le schéma suivant représente ces différentes relations entre la géométrie, les entités géométriques, les champs géométriques, le maillage et les champs éléments finis :

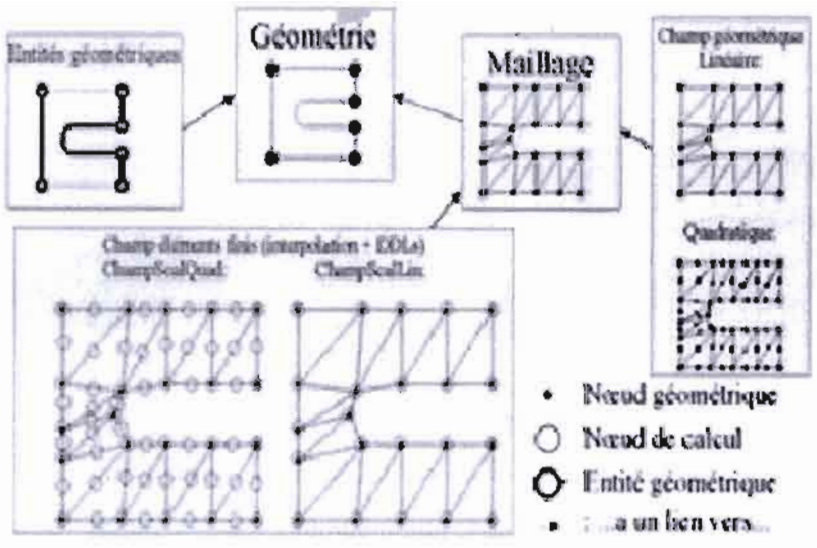


Fig. II.20: relations entre la géométrie, les entités, les maillages et les différents champs

Dans cette figure, les nœuds géométriques désignent les coordonnées des sommets et des milieux d'arêtes d'un maillage. .

V. 5 Formulation variationnelle et résolution d'un problème

Dans le document d'introduction à la méthode des éléments finis, on a vu que le système global est obtenu à partir de la formulation variationnelle du problème aux limites. Ainsi, l'usager désirant développer une nouvelle application doit en premier lieu écrire la formulation variationnelle du problème. Par conséquent, la formulation variationnelle va permettre d'écrire la matrice et le résidu élémentaire. Cette tâche peut être complexe surtout si le problème comporte plusieurs inconnues, c'est-à-dire les problèmes couplés.

V. 5. 1 Détails de la résolution d'un problème

Nous voici rendu à décrire en détail la résolution d'un problème dans le CCEF :

- Lecture des données d'entrée (les 5 fichiers typiques - .géométrie, champ, .CL, .matériaux et chargement)
- Affectation des champs

- Assemblage et résolution du système global
- Exportation des résultats

V. 5. 1. 1 Lecture des données

- Cette étape consiste simplement à lire les données sur disque. Pour résoudre un problème, on devra avoir les données suivantes :
- Géométrie (fichier .geom, classe « Geometrie »)
- Entités géométriques (fichier .enti, vecteur « m-geometrie »)
- Maillage (fichier .mail, classe « Maillage »)
- Les champs pour chaque variable du problème (fichier .champs, ex : classe « GestionFichierChamps »)
- Liste de conditions aux limites (fichier .CL, ex : classe « ListeConditionsLimites »)

Une fois ces données en mémoire, nous sommes prêts à les utiliser.

V. 5. 1. 2 Assemblage et résolution du système global

Pour former le système d'équations éléments finis, il faut d'abord numéroter les degrés de liberté de 1 à n, où n est le nombre total de degrés de liberté (DDL) d_l, y compris les DDL fixés d_f. Pour garantir la structure hiérarchique des matrices et faciliter l'assemblage et la condensation du système, la numérotation doit respecter deux critères.

- Critère 1 : Les DDL fixés doivent porter un numéro plus élevé que les DDL libres, ceci afin d'éliminer ces DDL fixés et de calculer les réactions.

- Critère 2 : Dans chacun des groupes de DDL (fixés ou libres), les DDL relatifs à un degré p doivent porter un numéro plus élevé que ceux relatifs aux degrés p-1, p-2,...,1,ceci afin de préserver la structure hiérarchique du système d'équations éléments finis.

Maintenant que nous avons un problème à résoudre, on peut demander à un solveur de l'utiliser pour faire l'assemblage de la matrice et du vecteur global et ensuite faire la résolution du système ainsi créé.

La méthode d'assemblage utilisée dans le moteur de calcul est un peu spéciale, elle diffère de la méthode décrite en (III.4) .

L'originalité de notre méthode réside dans la construction d'une matrice booléenne de permutation $[P]$ qui permet de mettre la raideur structurale $[K]$ sous la forme

$$[\mathrm{K}]=\begin{bmatrix} K_{ll} & K_{li} \\ K_{il} & K_{ii} \end{bmatrix}$$

et

les vecteurs des déplacements et chargements sous la forme

$$d=\begin{bmatrix} d_l \\ d_i \end{bmatrix} \text{ et } F=\begin{bmatrix} F_l \\ F_i \end{bmatrix}$$

La résolution du système linéaire suivant permet d'avoir les déplacements cherchés.

$$\begin{bmatrix} F_l \\ F_i \end{bmatrix}=\begin{bmatrix} K_{ll} & K_{li} \\ K_{il} & K_{ii} \end{bmatrix}\begin{bmatrix} d_l \\ d_i \end{bmatrix}$$

D'où on sort :

$$d_l = K_{ll}^{-1}(F_l - K_{li}d_i)$$

Une fois résolu, nous avons la solution du problème pour nos différentes inconnues.

V. 5. 1. 3 Exportation des résultats

La dernière étape consiste simplement à exporter les résultats calculés pour la visualisation. L'exportation se fera selon le visualisateur désiré : la fenêtre vue de notre application ou un éditeur de texte approprié.

V. 5. 1. 4 Post-traitement

Afin de permettre à l'utilisateur une analyse aisée de la solution éléments finis, il est nécessaire de calculer certaines grandeurs (déplacements, contraintes, ...) à partir du vecteur solution du système d'équations linéaires et de les afficher sous un format de sortie facilement interprétable.

Le post-traitement consiste en l'utilisation des résultats obtenus en déplacements pour le calcul des déformations et éventuellement des contraintes dans la structure par le biais des lois de comportement des matériaux.

Pour un matériau isotrope à comportement linéaire par exemple ,les déformations en tout point sont données par :

$$\left\{\varepsilon^e(x,y)\right\} = \left[L\right]\left\{U(x,y)\right\}$$

Où $\left[L\right]$ est l'opérateur défini de la façon suivante :

$$\left[L\right] = \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{bmatrix}$$

avec $\left[U(x,y)\right] = \left[N_e(x,y)\right]\left[U_e\right]$

d'où

$$\left\{\varepsilon^e(x,y)\right\} = \left[L\right]\left[N^e(x,y)\right]\left\{U_e\right\} = \left[B(x,y)\right]\left\{U_e\right\}$$

Et par la suite les contraintes sont obtenues par la relation suivante :

$$\left\{\sigma(x,y)\right\} = E\left\{\varepsilon^e(x,y)\right\}$$

CONCLUSION et RECOMMANDATIONS

La finalité de ce projet était de développer la méthode des éléments finis dans le cadre de l'analyse structurale. En effet la modélisation des structures élastiques aboutit à des équations différentielles dont la résolution analytique est souvent impossible mais en faisant appel à une formulation variationnelle basée sur le principe des travaux virtuelles et le minimum de l'énergie potentielle totale, nous arrivons à discrétiser le problème et à trouver une solution approchée nous permettant de résoudre le problème d'élasticité qui dans notre cas correspond à l'analyse structurale d'éléments surfaciques.

Cependant le développement de cette méthode étant fastidieuse, l'utilisation d'outils informatiques s'impose pour la mise en œuvre de cette dernière. La principale difficulté réside dans le choix du langage et des structures de données adéquates pour représenter les différentes grandeurs utilisées lors de l'application de la méthode. La recherche d'une certaine convivialité dans l'acquisition des données de base et dans l'affichage des résultats est aussi importante que la création d'une interface graphique adéquate, attrayante, constituant souvent un aspect non négligeable d'un programme utilisant la méthode des éléments finis.

Dans le cadre de l'analyse structurale la prise en compte de tous les types de sollicitations notamment celles dues aux forts gradients de contraintes, est primordiale. Ainsi de nos jours, l'analyse dynamique entre en jeu avec d'autres méthodes basées sur les éléments finis (méthode de Trefftz). La méthode de Trefftz pourrait ultérieurement faire l'objet de projet de fin d'études.

De plus comme toute méthode de discrétisation, il existe une erreur d'approximation dont la grandeur détermine la validité de la méthode utilisée. Cette erreur est due non seulement au principe de discrétisation utilisé mais aussi aux algorithmes informatiques qui génèrent des erreurs d'approximation ; des recherches peuvent être axée sur l'amélioration des algorithmes utilisés afin de minimiser les erreurs dues à la machine.

Nos dernières recommandations vont à l'encontre de l'enseignement qui doit intégrer dans le cours d'informatique, des modules pour la programmation orientée objet (POO).

BIBLIOGRAPHIE

[1] Dr Moustapha NDIAYE, La Méthode des éléments finis appliquée à l'analyse des structures et des solides,Cours de structures II , Ecole polytechnique de Thiès.

[2] Jean Charles CRAVEUR, Modélisation des structures, calcul par éléments finis.

[3] Desai ABEL , Introduction to the finite element method

[4] J.L. BATOZ et G. DHATT , Modélisation des structures par éléments finis, poutres et plaques.

[5] J.L. BATOZ et G. DHATT , Modélisation des structures par éléments finis, solides élastiques.

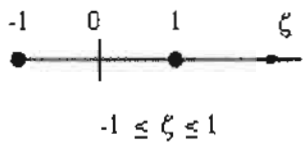
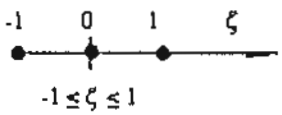
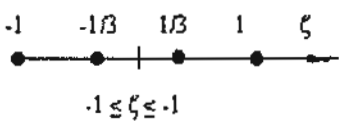
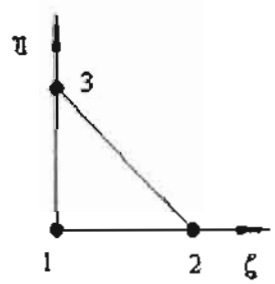
Sites internet :

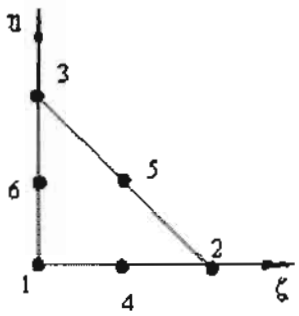
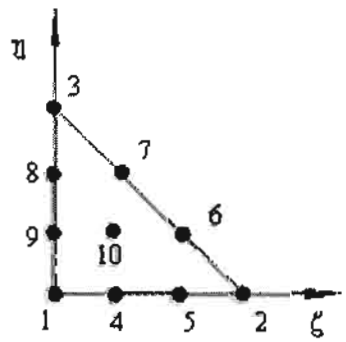
http:// www.**giref**.ulaval.ca

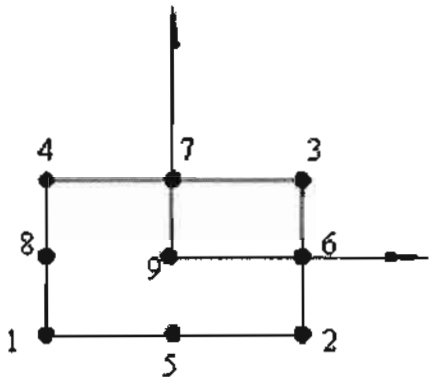
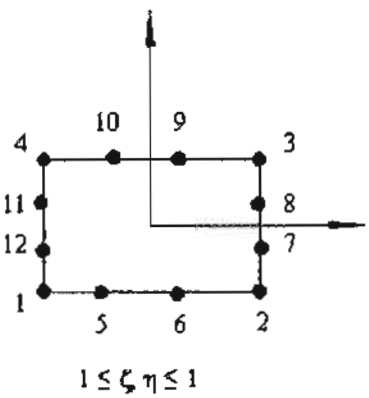
ANNEXES

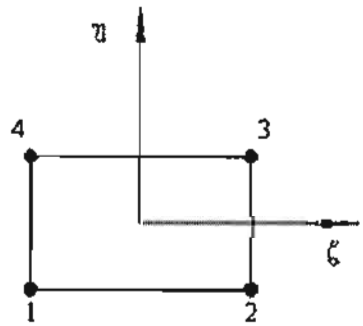
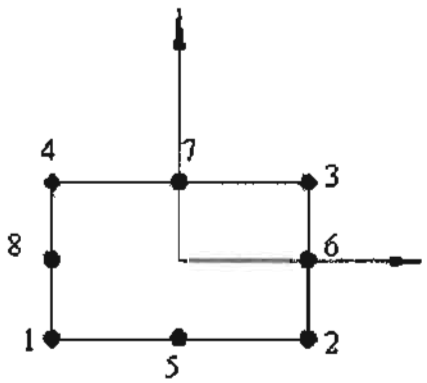
Bibliothèques d'éléments finis :

Nous donnons dans ce qui suit les éléments de références fréquemment utilisés ainsi que leurs fonctions d'interpolation.

Eléments de référence	Nombre de noeuds	Fonctions d'interpolation
 $-1 \le \zeta \le 1$	2	$N_1 = \frac{1}{2}(1-\zeta)$ $N_2 = \frac{1}{2}(1+\zeta)$
 $-1 \le \zeta \le 1$	3	$N_1 = \frac{1}{2}(-1+\zeta)\zeta$ $N_2 = \frac{1}{2}(1+\zeta)\zeta$ $N_3 = (1-\zeta)(1+\zeta)$
 $-1 \le \zeta \le -1$	4	$N_1 = \frac{1}{16}(1-\zeta)(3\zeta-1)(3\zeta+1)$ $N_2 = \frac{1}{16}(1+\zeta)(3\zeta-1)(3\zeta+1)$ $N_3 = \frac{9}{16}(1-\zeta)(1+\zeta)(1-3\zeta)$ $N_4 = \frac{9}{16}(1-\zeta)(1+\zeta)(3\zeta+1)$
 $-1 \le \zeta, \eta \le 1$	3	$N_1 = 1-\zeta-\eta$ $N_2 = \zeta$ $N_3 = \eta$

Eléments de référence	Nombre de noeuds	Fonctions d'interpolation
	6	$N_1 = 1-3\zeta-3\eta+2\zeta^2+8\zeta\eta+2\eta^2$ $N_2 = \zeta(-1+2\zeta)$ $N_3 = \eta(-1+2\eta)$ $N_4 = -4\zeta(-1+\zeta+2\eta)$ $N_5 = 4\zeta\eta$ $N_6 = -4\eta(-1+\zeta+\eta)$
	10	$N_1 = -\frac{1}{2}(3\eta-2+3\zeta)(-1+\zeta+\eta)(3\eta-1+3\zeta)$ $N_2 = \frac{1}{2}\zeta(-1+3\zeta)(3\zeta-2)$ $N_3 = \frac{1}{2}\eta(-1+3\eta)(3\eta-2)$ $N_4 = \frac{9}{2}\zeta(-2+3\eta+3\zeta)(-1+\zeta+\eta)$ $N_5 = -\frac{9}{2}\zeta(-1+3\zeta)(-1+\zeta+\eta)$ $N_6 = \frac{9}{2}\zeta\eta(-1+3\zeta)$ $N_7 = \frac{9}{2}\zeta\eta(-1+3\eta)$ $N_8 = -\frac{9}{2}\eta(-1+3\eta)(-1+\zeta+\eta)$ $N_9 = \frac{9}{2}\eta(3\eta-2+3\zeta)(-1+\zeta+\eta)$ $N_{10} = -27\zeta\eta(-1+\zeta+\eta)$

Eléments de référence	Nombre de noeuds	Fonctions d'interpolation
 $-1 \le \zeta\ \eta \le 1$	9	$N_1 = \frac{1}{4}\zeta(-1+\eta)(2\zeta\eta+\zeta+1)$ $N_2 = -\frac{1}{4}\zeta(-1+\eta)(1+\zeta)$ $N_3 = \frac{1}{4}\zeta(1+\eta)(2\eta-1)(1+\zeta)$ $N_4 = \frac{1}{4}\zeta(1+\eta)(1+\zeta-2\eta)$ $N_5 = -\frac{1}{2}(-1+\eta)(1+\zeta)(\zeta-\eta+2\zeta\eta)$ $N_6 = -\zeta(-1+\eta)(1+\eta)(1+\zeta)$ $N_7 = -\frac{1}{2}(1+\eta)(1+\zeta)(\zeta-\eta)$ $N_8 = -\zeta^2(-1+\eta)(1+\eta)$ $N_9 = (-1+\eta)(1+\eta)(1+\zeta)(2\zeta-1)$
 $1 \le \zeta\ \eta \le 1$	12	$N_i = \frac{1}{32}(1-\zeta\zeta_i)(1+\eta\eta_i)(-10+9\zeta^2+9\eta^2)$ i = 1, 2, 3 ,4 $N_i = \frac{9}{32}(1+\zeta\zeta_i)(1+9\eta\eta_i)(1-\eta^2)$ i = 7, 8, 11, 12 $N_i = \frac{9}{32}(1+9\zeta\zeta_i)(1+\eta\eta_i)(1-\zeta^2)$ i = 5, 6, 9, 10

Eléments de référence	Nombre de noeuds	Fonctions d'interpolation
 $-1 \le \zeta, \eta \le 1$	4	$N_1 = \frac{1}{4}(1-\zeta)(1-\eta)$ $N_2 = \frac{1}{4}(1+\zeta)(1-\eta)$ $N_3 = \frac{1}{4}(1+\zeta)(1+\eta)$ $N_4 = \frac{1}{4}(1-\zeta)(1+\eta)$
 $-1 \le \zeta, \eta \le 1$	8	$N_1 = \frac{1}{4}(1-\zeta-\eta)(1-\eta)(1-\zeta)$ $N_2 = \frac{1}{4}(-1+\zeta-\eta)(1-\eta)(1+\zeta)$ $N_3 = \frac{1}{4}(-1+\zeta+\eta)(1+\eta)(1+\zeta)$ $N_4 = \frac{1}{4}(-1-\zeta-\eta)(1+\eta)(1-\zeta)$ $N_5 = \frac{1}{2}(1-\eta)(1-\zeta^2)$ $N_7 = \frac{1}{2}(1+\eta)(1-\zeta^2)$ $N_8 = \frac{1}{2}(1-\eta^2)(1-\zeta)$

Algorithme de maillage d'une plaque rectangulaire

Cet algorithme effectue le maillage d'une plaque dont les dimensions L et l sont connues. On subdivise ainsi la plaque en plusieurs éléments délimités par des nœuds dont les coordonnées sont stockées dans un tableau nommé nœud[]. Les nombres nx et ny donnant le nombre d'éléments suivant x et suivant y respectivement.

Données d'entrée :

- L : longueur de la plaque
- l : largeur de la plaque
- nx : nombre d'éléments suivant l'axe x
- ny : nombre d'éléments suivant y

Données de sortie

- nœud [] : tableau d'enregistrement des coordonnées des nœuds de la plaque
- n : nombre total de noeuds
- dx : pas de la discretisation suivant x
- dy : pas de la discretisation suivant y

Algorithme :

```
Début
    dx ⟵ L/nx ;
    dy ⟵ l/ny ;
    rx ⟵ L mod nx ;  \\ reste de la division de L par nx
    ry ⟵ l mod ny ;  \\ reste de la division de l par ny

j=0 ; n=0  // initialisation

tant que (j<=ny) faire
début tant que
    pour i allant de 1 à nx faire
    début pour i
        nœud[n].x ⟵ i*dx ;
        nœud[n].y ⟵ i*dy ;
```

```
            si (rx ≠ 0) et (i = nx) alors
            debut si
                  n ←— n+1 ;
                  nœud[n].x ←— L ;
                  nœud[n].y ←— j*dy ;
            fin si
            n ←— n+1 ;
      fin pour i

      si (ry ≠ 0) et (j = ny) alors
      début si
            pour i allant de 1 à nx faire
            début pour i
                  nœud[n].x ←— i*dx ;
                  nœud[n].y ←— i*dy ;
            si (rx ≠ 0) et (i = nx) alors
            debut si
                  n ←— n+1 ;
                  nœud[n].x ←— L ;
                  nœud[n].y ←— l ;
            fin si
            n ←— n+1 ;
      fin pour i
      j ←— j +1 ;
fin tant que

FIN.
```

Au sortir de l'algorithme nous avons un tableau de n nœuds ainsi que leurs coordonnées.

Algorithme de création d'un tableau de connectivité

Cet algorithme crée un tableau nomme maille, contenant un classement selon le numéro de tous les mailles d'une plaque rectangulaire dont le maillage à été fait au préalable. Nous connaissons donc les numéros des nœuds et le nombre total de mailles. Chaque élément du tableau est un enregistrement dont les champs sont les numéros des quatre nœuds de chaque maille.

Données d'entrée :

- nx : nombre de mailles suivant l'axe x
- ny : nombre d'éléments suivant y

Données de sortie :

-maille[] : tableau de connectivité de la plaque

Algorithme

```
Début
    pour i_rangée allant de 1 à ny faire
    début pour_ i_rangée
        pour i allant de 1 à nx faire
        début pour_i
            indice ← nx*(i_rangée − 1) + i ;
            maille[indice].prem ← i_rangée + indice -1 ;
                maille[indice].deux ← maille[indice].prem + 1 ;
                maille[indice].trois ← maille[indice].deux + nx + 1 ;
                maille[indice].quatre ← maille[indice].trois − 1 ;
        fin pour_i
    fin pour i_rangée
FIN
```

$$[K]=\begin{bmatrix}
\frac{4H_1}{a^2}+\frac{4G}{b^2} & \frac{3c}{ab} & \frac{-4H_1}{a^2}+\frac{2G}{b^2} & \frac{3d}{ab} & \frac{-2H_1}{a^2}-\frac{2G}{b^2} & \frac{-3c}{ab} & \frac{2H_1}{a^2}-\frac{4G}{b^2} & \frac{-3d}{ab} \\
 & \frac{4H_1}{b^2}+\frac{4G}{a^2} & \frac{-3d}{ab} & \frac{2H_1}{b^2}-\frac{4G}{a^2} & \frac{-3c}{ab} & \frac{-2H_1}{b^2}-\frac{2G}{a^2} & \frac{3d}{ab} & \frac{-4H_1}{b^2}-\frac{2G}{a^2} \\
 & & \frac{4H_1}{a^2}+\frac{4G}{b^2} & \frac{-3c}{ab} & \frac{2H_1}{a^2}-\frac{4G}{b^2} & \frac{3d}{ab} & \frac{-2H_1}{a^2}-\frac{2G}{b^2} & \frac{3c}{ab} \\
 & & & \frac{4H_1}{b^2}+\frac{4G}{a^2} & \frac{-3d}{ab} & \frac{-4H_1}{b^2}+\frac{2G}{a^2} & \frac{3c}{ab} & \frac{-2H_1}{b^2}-\frac{2G}{a^2} \\
 & & & & \frac{4H_1}{a^2}+\frac{4G}{b^2} & \frac{3c}{ab} & \frac{-4H_1}{a^2}+\frac{2G}{b^2} & \frac{3d}{ab} \\
 & & & & & \frac{4H_1}{b^2}+\frac{4G}{a^2} & \frac{-3d}{ab} & \frac{2H_1}{b^2}-\frac{4G}{a^2} \\
 & & & & & & \frac{4H_1}{a^2}+\frac{4G}{b^2} & \frac{-3c}{ab} \\
 & & \text{SYM} & & & & & \frac{4H_1}{b^2}+\frac{4G}{a^2}
\end{bmatrix}$$

Matrice de rigidité symétrique d'un rectangle à huit dll (matériau isotrope)

Avec :

$$\boxed{G=\frac{E}{2(1+\nu)}} \qquad \boxed{H_1=\frac{2G(1-\alpha\nu)}{(1-\nu-\alpha\nu)}}; \qquad \boxed{H_2=\frac{\nu H_1}{1-\alpha\nu}}$$

Où $\alpha=0$ en contrainte plane et $\alpha=1$ en déformation plane.

$$\boxed{c=H_2+G};$$

$$\boxed{d=H_2-G};$$