Étude et réalisation de logiciels d'éléments finis sur micro-ordinateur : mailleurs automatiques résolution auto-adaptative de problèmes de mécanique utilisant la méthode multigrilles

Christian Le Carlier de Veslud

INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

FORMATION DOCTORALE DE MÉCANIQUE-ENERGÉTIQUE

ETUDE ET REALISATION DE LOGICIELS D'ELEMENTS FINIS SUR MICRO ORDINATEUR : MAILLEURS AUTOMATIQUES, RESOLUTION AUTO-ADAPTATIVE DE PROBLEMES DE MECANIQUES UTILISANT LES METHODES MULTIGRILLES

THESE

présentée devant l'Institut National Polytechnique de Lorraine
pour l'obtention du titre de

DOCTEUR de l' I.N.P.L

(Mécanique-Energétique)

par

Christian Le Carlier de Veslud

Soutenance prévue le 12 novembre 1991 devant la commission d'examen

JURY

Monsieur le professeur	M. GANTOIS	Président
Monsieur le professeur	M. PIERRE	} Rapporteurs
Monsieur le professeur	O. PIRONNEAU	
Madame le professeur	H. DUCAUQUIS	} Examinateurs
Monsieur le professeur	M. GUEURY	
Monsieur le professeur	G. MAURICE	Directeur

Résumé

Le travail présenté ici consiste en l'étude et le développement d'un ensemble de logiciels sur micro-ordinateur Macintosh pour la simulation numérique de problèmes de mécanique utilisant les méthodes multigrilles. On peut y distinguer deux parties:

- la première consacrée aux méthodes de maillage automatique
- la seconde traitant des applications mécaniques: la simulation de problèmes d'ossatures planes linéaires de poutres et d'élasticité linéaire plane.

Les difficultés essentielles du projet consistent, d'une part à s'adapter aux possibilités limitées du micro-ordinateur pour traiter des problèmes qui peuvent être très vite de taille importante, et d'autre part, de garantir la fiabilité du logiciel tout en offrant à l'utilisateur simplicité d'emploi et automaticité utilisant au maximum la convivialité de l'appareil.

La première partie concerne la réalisation de logiciels de maillage automatiques en 2D et 3D. Deux logiciels sont présentés dans chaque cas. Un premier choix offre un maillage par blocs grossiers 2D ou 3D définis par une transformation, puis maillés séparément couche par couche. Un deuxième choix 2D est la méthode de Delaunay qui permet de construire automatiquement une triangulation à partir du contour. Enfin, un mailleur 3D par élévation cylindrique de maillages 2D est mis en place. Divers utilitaires y sont adjoints: modifications par suppressions ou adjonctions d'entités, reproduction par transformations géométriques simples, visualisation et calcul des parties cachées...etc.

La seconde partie est consacrée au développement systématique des méthodes multigrilles sur micro-ordinateur. Nous présentons une structure de données spécifique associée à la méthode Full-multigrilles ainsi que des processus de raffinement automatique. Les outils précédemment développés sont utilisés à deux types d'applications. D'abord, le calcul de la déformation d'ossatures planes linéaires de poutres avec ou sans cisaillement transverse. Enfin, une application à l'élasticité plane bidimensionnelle est présentée. Cette dernière est complétée par un processus de raffinement auto-adaptatif du maillage tenant compte des erreurs locales. Ce qui permet au logiciel de concentrer ses efforts là où c'est nécessaire, économisant ainsi place mémoire et temps de calcul, une optique particulièrement bien adaptée au cas des micro-ordinateurs.

Remerciements

Ce travail, commencé dans le Laboratoire d'Energétique et de Mécanique Théorique et Appliquée (LEMTA), s'est terminé dans le Laboratoire des Sciences et Génies des Surfaces (LSGS). Il a plus précisément été réalisé sous la direction de M le Professeur G. Maurice et dans le cadre du groupe de Modélisation théorique et numérique qu'il anime. Je tiens à le remercier pour l'aide, les conseils et les critiques qu'il m'a toujours apporté dans l'élaboration de ma thèse.

Je suis honoré que M le Professeur M. Gantois, Président de l'INPL et Directeur du LSGS ait accepté de présider le jury de cette thèse malgré ses lourdes charges.

Je remercie M O. Pironneau, Professeur à l'université de Paris-VI, et Chef de projet à l'INRIA-Rocquencourt, ainsi que M M. Pierre, Professeur à l'université de Nancy-I et Chef de projet à l'INRIA-Nancy d'avoir bien voulu accepter de rapporter mes sur travaux et de m'éclairer de leurs conseils.

Mme le Professeur H. Ducauquis, responsable de la formation doctorale à l'INPL, nous a fait l'honneur de participer au jury; qu'elle en soit vivement remerciée.

Je remercie M Le professeur M. Gueury, directeur de l'ERIN (Equipe de Recherche en Interfaces Numériques) de nous avoir fait l'honneur de s'intéresser à nos travaux et de participer à notre jury.

Un grand merci aux camarades du groupe qui ont su maintenir dans nos bureaux une ambiance sympathique.

Je remercie le personnel du LEMTA et de l'ENSEM (Ecole Nationale Supérieure d'Electricité et de Mécanique) qui m'a apporté une aide précieuse, et particulièrement Mmes C. Ducarne, M.H.Zobermann et O. Iung.

Je n'oublie pas non plus mes camarades chercheurs au LEMTA pour la bonne ambiance qu'ils ont su faire régner dans les locaux.

Je remercie M le Professeur J.C Braun, Directeur de l'ENSEM pour l'aide qu'il m'a apportée.

Le personnel du LSGC m'a apporté une aide précieuse; je tient à l'en remercier, et plus particulièrement Mmes Lehmann et Colinet.

Enfin, je tiens à remercier les élèves de l'ENSEM de toutes les promotions pour la patience avec laquelle ils m'ont supporté et pour le soutien moral qu'ils m'ont apporté.

A ma famille

A ma Grand-Mère

A mon Sethi

Symboles et Notations

1-Symboles

[ ]	Matrice
< >	Vecteut lignz
{ }	Vecteur colonne
(u,v)	Produit scalaire de u par v
$f_{i,j}$	Dérivée partielle de f_i par x_j

2-Notations

E	Module d'Young du matériau
ν	Coefficient de Poisson
G	Module de Coulomb
ρ	Masse volumique du matériau
A	Aire de chaque poutre
I	Inertie principale de flexion de chaque poutre
m	Moment statique
$\Gamma(0)$	Configuration initiale du corps
$\Gamma(t)$	Configuration du corps à l'instant t
σ	Tenseur des contraintes de Cauchy-Euler
$\sigma_{i,j}$	Composante du tenseur des contraintes de Cauchy-Euler
ε	Tenseur des déformations infiniment petites
$\varepsilon_{i,j}$	Composante du tenseur des déformations infiniment petites
$\delta_{i,j}$	Symbole de kronecker
h	Taille de la grille (de la grille fine si il en a deux)
H	Taille de la grille grossière
h_i	Taille de la grille N° i (dans le cas de plusieurs grilles)
Ω	Domaine d'étude
Γ	Contour externe du domaine Ω
Ω_h	Discrétisation du domaine Ω par une grille de taille h
Ω_i	Elément fini N°i de la grille Ω_h
S	Opérateur de lissage

u ou u_{ex}	Solution théorique
u_h	Solution approchée par discrétisation sur la grille Ω_h
$\hat{v}$, $\tilde{v}$	Valeurs approchées de v
e_h	Erreur sur la grille Ω_h
d_h	Défaut sur la grille Ω_h
dW	Travail élémentaire
$[K]$	Matrice de rigidité
$[D]$	Matrice d'élasticité
$[M]$	Martrice de masse cohérente
$[B]$	Matrice reliant les déformations aux valeurs nodales
$\{f\}$	Vecteur du chargement
N_i	Fonction de forme N° i
(u,v,w)	Déplacements suivant les axes ox, oy, oz
a_i	Valeur de la variable a au nœud N° i
θ	Rotation autour de l'axe oy (poutre)
β	Déformation de cisaillement transverse (poutre)
α	rotation de la section normale à la fibre moyenne (poutre)
k	Coefficient de correction de cisaillement (poutre)
τ	Contrainte de cisaillement transverse (poutre)
N	Effort normal
T	Effort tranchant
$\underline{M}$	Moment fléchissant
σ	Contrainte moyennée aux nœuds

Table des matières

Chapitre IV : Maillage en dimension 3

PARTIE B : Résolution par méthode multigrilles

Chapitre V : Méthodes multigrilles

Chapitre VI : Structure de données multigrilles

Chapitre VII : Structure planes formées d'assemblage de poutres

Introduction

Dans ce mémoire, nous présentons l'étude et la réalisation d'un ensemble de logiciels permettant la modélisation de problèmes de mécanique par éléments finis et leur résolution par des méthodes multigrilles sur micro-ordinateur Macintosh.

Ce travail a été réalisé dans le groupe de modélisation théorique et numérique que dirige M le professeur G. Maurice. Ce groupe poursuit depuis de nombreuses années des travaux de recherche sur la modélisation et la résolution théorique et numérique de problèmes mécaniques appliqués du domaine de l'ingénieur. Les activités du groupe s'exercent dans le domaine de la mécanique des fluides, de la mécanique des solides et des méthodes numériques nouvelles. Beaucoup de ces études ont débouché sur des applications industrielles ou médicales.

Avec l'avènement des ordinateurs et le développement rapide de ceux-ci, la méthode des éléments finis a connu un impressionnant développement. Cette méthode couvre des domaines très variés de la Mécanique, de la Thermique, de la Electricité mais aussi de la Biologie, et de la Médecine. Elle est donc de plus en plus employée dans de nombreux laboratoires et bureaux d'étude.

Le domaine particulier de la mécanique des solides fait l'objet de nombreuses applications industrielles, notamment pour le calcul d'optimisation de structures, problèmes courant dans l'aéronautique, la mécanique automobile ou l'aérospatiale. Conjointement à cette utilisation croissante, de nombreuses recherches théoriques et appliquées, principalement à partir des travaux de Zienkiewicz, ont permis de développer des modèles de plus en plus performants.

Actuellement, les problèmes linéaires sont assez bien maîtrisés. Par contre, la modélisation de problèmes plus complexes conduisant à des non-linéarités est encore incertaine et fait l'objet de nombreuses recherches.

Les progrès fulgurant réalisés par les micro-ordinateurs ont maintenant rendu l'utilisation de la méthode accessibles à ceux qui ne disposent pas de gros moyens de calculs. En effet, le micro-ordinateur est un outil souple, convivial, indépendant et surtout

peu onéreux (tant au niveau de l'achat qu'à celui de l'entretien). Cependant, ce genre d'outil présente deux inconvénients majeurs : une place mémoire limitée et des temps de calculs assez longs.

Une adaptation judicieuse de la méthode des éléments finis sur micro-ordinateur nécessite donc un choix judicieux des méthodes les plus fines et les plus performantes de l'analyse numérique pour pouvoir répondre à ces deux impératifs.

C'est dans cette optique que nous nous sommes intéressés à l'utilisation et au développement des méthodes multigrilles. Ces méthodes permettent d'améliorer la rapidité de résolution des problèmes issus de la modélisation par éléments finis ou différences finies, tout particulièrement pour les problèmes de grosses tailles. Apparues dans les années 1960 en URSS, leur efficacité a été initialement prouvée sur quelques problèmes tests (essentiellement, l'équation de Poisson). Vers les années 1970, l'école Allemande avec Hackbush et Trottenberg a repris l'étude des méthodes multigrilles de façon systématique, ce qui a conduit à une théorie unifiée. Le flambeau de l'étude a été repris dans de nombreux pays, en France, notamment par J.F Maitre et plus près de nous par le laboratoire de M. Pierre.

Cette méthode a surtout été appliquée avec succès en mécanique des fluides, et à une moindre échelle, en mécanique des solides. Cependant, le comportement théorique exact de la méthode, particulièrement dans le cas des éléments finis, n'a été explicité que dans quelques cas simples (essentiellement des problèmes elliptiques). Le comportement général de la méthode dans un cas quelconque reste encore mal connu. Et aujourd'hui encore, de nombreuse recherches sont en cours pour dominer les méthode multigrilles dans un cas général, particulièrement dans le cas de problèmes hyperboliques.

Comme nous l'avons déjà mentionnés, la méthode multigrilles est utilisée dans le domaine des éléments finis, essentiellement sur gros ordinateur et souvent en adaptant des codes prévus pour des méthodes classiques. Nous avons, pour notre part, décidé de reprendre le problème à la base et de re-formuler des principes de développement systématiques des méthodes multigrilles en éléments finis. Nous avons étudié parmi les algorithmes et les structures de données, ceux qui répondaient le mieux aux exigences (parfois contradictoires) que nous nous étions imposées (place mémoire, temps de calcul, stockage des données etc...). Le résultat de cette étude comparative nous permet de proposer des principes de mise en œuvre de la méthode, mais aussi d'en déterminer les limites d'application pour notre étude.

Ces principes ont été mis en application en 1D (ossatures planes linéaires de poutres avec formulation de Midlin ou d'Euler Bernouilli incluant ou non le cisaillement transverse) et en 2D (élasticité plane linéaire). Particulièrement dans ce dernier cas, nous proposons une méthode de résolution par méthode multigrilles sur des maillages auto-adaptatifs. Cette variante, quoique complexe, est particulièrement bien adaptée au cas des micro-ordinateurs. Elle permet au programme de calcul de concentrer ses efforts uniquement là où c'est nécessaire. Il s'ne suit un gain notable de place mémoire et de temps de calcul.

Il apparait donc que le développement des méthodes multigrilles constitue un tout (assez complexe d'ailleurs) dans lequel résolution et maillage sont fortement liés. Pour assurer une mise ne œuvre efficace, nous avons donc décidé de développer un "environnement éléments finis multigrilles" permettant de valoriser la méthode de résolution utilisée.

Nous avons tout particulièrement concentré nos efforts sur l'étude et la réalisation de mailleurs automatiques. Nous les avons utilisé pour générer aisément des grilles de départ grossières mais aussi pour reconstruire les grilles fines intervenant lors du maillage auto-adaptatif.

Les méthodes de réalisation automatique de maillages posent un grand nombre de problèmes dûs au fait qu'il y a peu d'algorithmes généraux. La diversité géométrique des types d'éléments finis (tant en 2D, qu'en 3D) complique encore le problème. Nous proposons donc d'abord une structure de données complète inspirée de celles utilisées sur les gros ordinateurs et conduisant description précise de maillages. En nous appuyant sur cette base, nous avons ensuite élaborés quatre algorithmes de maillages automatiques ou semi-automatiques en 2D ou en 3D.

Nous avons mis l'accent sur l'automaticité et la facilité d'emploi des logiciels de maillages crées. Les stratégies utilisées permettant à l'utilisateur de construire des domaines complexes bi- et tri-dimensionnels aisément.

D'un point de vue purement informatique, l'architecture de nos logiciels répond à plusieurs impératifs:

- Une programmation modulaire permettant de modifier facilement le programme ou de développer d'autres applications.

- Une gestion de données et des résultats aisée utilisant au maximum les possibilités graphiques du Macintosh.

Pour répondre à cela, nous avons programmé nos logiciels en Fortran 77. L'interface graphique (toujours écrite en Fortran) utilise la Toolbox Macintosh

(bibliothèque graphique). Grâce à cela, les logiciels sont gérés par menus déroulants, l'acquisition des données comme l'exploitation des résultats se fait par saisie directe à l'écran. De plus, un grand nombre d'utilitaires (tant en maillage qu'en résolution) ont été inclus pour faciliter le travail de l'utilisateur et augmenter l'efficacité de nos logiciels. Nous avons en cela voulu suivre la voie tracée par le logiciel MacFEM d'O. Pironneau, qui fut l'un des premiers logiciels d'éléments finis sur Macintosh vraiment convivial.

Les données d'entrée ainsi que les résultats sont délocalisés sur des fichiers disques accessibles directement, ceci permet de gagner de la place mémoire et de pouvoir reprendre facilement des études.

Le mémoire que nous présentons se décompose en deux parties de quatres chapitres chacun. La première (chapitres I à IV) traite des problèmes de maillages, la seconde (chapitres V à VIII) de la résolution par méthodes multigrilles.

Le **chapitre I** décrit la structure de données utilisée pour décrire un maillage. Cette structure est importante car c'est sur elle que reposent à la fois nos phases de maillages et nos procédures de résolutions.

Le **chapitre II** aborde une première méthode de maillage bidimentionnelle : le maillage par blocs. Celle-ci consiste à décomposer le domaine à mailler en une série de sous-domaines de géométrie fixée. Chaque bloc est ensuite maillé séparément, puis l'ensemble est rassemblé pour fournir le maillage souhaité. L'entrée des données a été facilitée par l'incorporation de fonctions de DAO automatiques. Le logiciel offre une bibliothèque d'éléments assez étendue (triangles et quadrilatères de divers degrés). Le maillage final peut être facilement modifié ou complété par un arsenal d'utilitaires automatiques (symétries, rotations,...,mais aussi destructions ou modifications locales de points ou d'éléments). A chaque étape du programme, la structure de données est vérifiée et corrigée automatiquement de façon à être compatible avec les règles énoncées dans le chapitre I.

Le **chapitre III** étudie la réalisation d'un logiciel de maillage cette fois-ci totalement automatique utilisant une méthode de Delaunay. L'utilisateur ne doit définir que le contour extérieur, tâche facilitée par des utilitaires de DAO. A partir de là, le programme maille automatiquement en triangles le domaine défini en assurant que le maillage construit respecte bien le contour fixé. Cette dernière étape est effectué par une méthode récente de "forçage d'arête" permettant, à partir d'un maillage donné, de construire un maillage équivalent respectant un contour fixé, et ce sans ajout de points. L'ensemble du maillage peut être modifié, comme dans le chapitre II par des post-traitements. En particulier, nous présentons un processus de remaillage en quadrilatères par regroupement de triangles ainsi qu'un processus d'ajout de points permettant d'augmenter le degrés d'interpolation des éléments crées.

Le **chapitre IV** est consacré au maillage tridimensionnel. Nous proposons deux méthodes de maillage. La première est une méthode "par bloc" similaire à celle développée au chapitre II. La seconde est une méthode "par couches" qui construit un maillage 3D par le déplacement d'un maillage plan dans l'espace selon un chemin fixé. La structure de données utilisée fait l'objet d'une étude particulière. La définition du domaine à mailler, particulièrement délicate en 3D, est réalisée à l'écran. La représentation du maillage dans l'espace nous conduit à développer un calcul des parties cachées, fournissant une représentation solide du domaine, option très souvent indispensable pour bien comprendre les formes complexes.

Avec le **chapitre V**, débute la partie consacrée à la résolution à proprement parler. Ce chapitre présente la théorie et l'application des méthodes multigrilles sur quelques exemples simples. L'étude de ceux ci met en évidence des tendances de comportement de la méthode qui sont très utiles pour le développement d'un code de calcul général. Il ressort de nos recherches que la méthode Full multigrilles est la plus efficace, c'est donc celle que nous avons choisi de mettre en œuvre.

Ainsi, après avoir choisi le type de méthode multigrilles le plus adapté, nous présentons dans le **chapitre VI** les algorithmes et las structures de données permettant d'appliquer le plus efficacement possible celle-ci sur micro-ordinateur. La méthode Full multigrilles développée utilise une définition dynamique des grilles permettant de traiter le problème de façon efficace. Le programme utilise un lissage par un splitting de Gauss-Seidel, l'équation de correction est résolue de façon exacte par un algorithme de gradient conjugué pré-conditionné. Le choix de la structure Morse (stockage de la matrice) permet de ne pas avoir à se soucier de la numérotation des nœuds et offre ainsi une plus grande souplesse d'utilisation entre les grilles et un gain de place mémoire.

Dans le **chapitre VII**, nous appliquons la méthode multigrilles présentée aux ossatures planes linéaires de poutres. Nous utilisons de éléments de Bernouilli (Hermitte de degrés 3) ou de Timoshenko (Lagrange de degrés 1,2 ou 3) permettant la prise en compte ou non du cisaillement transverse. Après la résolution, nous calculons puis nous présentons à l'écran les diagrammes de résistance des matériaux. Le logiciel est validé par une étude classique de poutres sur appuis.

Enfin, le **chapitre VIII** nous permet d'aborder le problème de l'élasticité plane. Nous proposons un algorithme de maillage auto-adaptatif bien adapté au méthodes multigrilles sur micro-ordinateur. La mise en œuvre d'une telle stratégie a nécessité l'élaboration de processus de remaillage permettant une définition compatible au sens des éléments finis. Nous validons à la fois l'algorithme et la stratégie de maillage par des essais classiques de mécanique de solides.

L'ensemble de ce travail a fait l'objet d'une publication dans le journal "Computational Mechanics Software for Engineering Workstations" en 1992.

L'architecture modulaire de ces logiciels et les nombreux utilitaires de maillage et de représentation ont permis de les utiliser dans d'autres applications, entre autres en mécanique des fluides, en homogénéisation. Il est par ailleurs prévu de développer d'autre applications, notamment dans le domaine tridimensionnel.

PARTIE A

Maillages automatiques en dimension 2 et 3

PARTIE A : Introduction

Principes de maillages automatiques

Malgré les récents développements du calcul formel, on ne peut espérer pouvoir disposer d'un outil général de résolution. Aussi, dans la plus grande majorité des cas, la méthode des éléments finis est actuellement de plus en plus fréquemment utilisée, tant en industrie que dans les laboratoires pour la simulation numérique des problèmes physiques très variés. Les progrès réalisés d'une part dans la modélisation et d'autre part dans la rapidité de la résolution (progrès numérique ou informatique) permettent de résoudre les problèmes physiques et géométriques les plus complexes.

Cependant, cette méthode exige dans un premier temps, une description géométrique précise du domaine à étudier par l'intermédiaire d'un maillage. C'est là une tache difficile et quelque fois périlleuse dont la difficulté est croissante avec la complexité du domaine et le nombre d'éléments souhaités.

Il a donc été conçu, de concert avec le développement de la méthode des éléments finis, de nombreux algorithmes de maillage bi- et tri-dimensionnels.

On peut globalement distinguer deux types de méthodes :

- la première consiste à considérer le domaine comme une transformation convenable d'un autre domaine simple à mailler.

- la seconde consiste à construire directement le maillage du domaine élément par élément, généralement à partir de la donnée de son contour.

Pour la dimension 2, citons les méthodes de Gordon, Winslow et Cook parmi celles du premier type, et les méthodes de Voronoï, Delaunay et George parmi celles du second type. Actuellement, en ce qui concerne le maillage en triangles, il existe des mailleurs totalement automatiques (du second type) traitant tous les cas possibles de façon satisfaisante. Pour le maillage en éléments quadrilatères, le problème est plus délicat et, à notre connaissance, il n'est pas résolu de façon totalement automatique.

En dimension 3, la plupart de ces méthodes sont encore applicables. Cependant, la plus grande complexité des domaines à mailler limite pour le moment leur application aux cas simples. Pour les géométries compliquées, seules les méthodes du second type sont utilisables, quoique peu d'entre elles donnent entièrement satisfaction. La complexité du problème se trouve aggravée par la difficulté qu'il y a à définir (analytiquement ou numériquement) un volume quelconque dans l'espace (sauf pour des géométries très simples) , définition indispensable ne serait-ce que pour décrire la peau du domaine. A cet égard, l'utilisation des méthodes de la CAO (essentiellement , les fonctions Splines ou les fonctions de Bézier) devraient permettre d'apporter une contribution importante à la représentation des volumes , en association avec la méthodes des éléments finis.

Les progrès réalisés depuis 10 ans en micro-informatique et le développement théoriques de méthodes très efficaces permettent maintenant l'utilisation d'algorithmes jusque là réservés aux gros systèmes.

Dans cette partie, nous présenterons diverses méthodes de maillage bi- et tri-dimensionnels dans l'optique particulière de leur utilisation sur micro-ordinateur. Pour chaque méthode, nous mettrons l'accent sur la structure de données nécessaire à la définition du problème ainsi qu'à la mémorisation du résultat obtenu. Nous traiterons aussi des divers avantages et inconvénients ainsi que des performances comparées de chaque méthode.

Chapitre I

Généralités sur les définitions de maillage

I-1 : Introduction

Ce chapitre a pour but de mettre en évidence les informations nécessaires à la constitution d'un maillage. Pour cela, nous examinerons les exigences auxquelles doit répondre un maillage. A partir de là, nous en déduirons les informations nécessaires à sa réalisation ainsi qu'un algorithme général de déroulement du processus.

I-2 : Définition d'un maillage

Un maillage est un ensemble de noeuds et d'éléments. Un noeud est un point géométrique supportant une ou plusieurs inconnues, un élément est un ensemble de noeuds associés à un domaine géométriquement simple La définition de chaque élément par les noeuds le composant est assujettie à des règles précises assurant sa validité (par exemple, en dimension 2, parcours des noeuds dans le sens trigonométrique direct)

Notons par Ω un domaine et par E un maillage formé de l'union de n éléments (e_i) i=1, ..,n construits sur p noeuds (P_i). Nous dirons que le maillage E décrit correctement le domaine Ω si l'on a :

$$\Omega = \bigcup_{i=1}^{n} e_i$$

- L'intersection de deux éléments quelconques est réduite à un point, une arête ou une face, selon la dimension du problème

- Tous les noeuds du maillage doivent appartenir à au moins un élément

A ces deux critères qui caractérisent la viabilité même du maillage s'ajoute un critère de "qualité" :

- les éléments doivent être aussi "réguliers" que possible.En dimension 2, par exemple, les triangles doivent se rapprocher du triangle équilatéral, et les quadrilatères du carré. Eventuellement, la qualité du maillage peut être améliorée par certaines opérations de régularisation.

Cette notion de "qualité" est importante car elle influe beaucoup sur la convergence des calculs. Par exemple, prenons deux maillages de même taille (même nombre de noeuds, même nombre d'éléments), l'un de bonne qualité, l'autre de mauvaise. Bien que conduisant à un problème de même taille, par rapport à la solution analytique, l'utilisation de l'un donnera de bien meilleurs résultats que l'utilisation de l'autre [EL 1]

En outre, la densité des éléments doit pouvoir varier selon le choix de l'utilisateur. Eventuellement, des éléments de type différents peuvent être utilisés, mais ils doivent être compatibles entre eux.

I-3 : Exigences a posteriori d'un maillage :

Une fois le maillage réalisé, la structure de données doit répondre aux exigences suivantes :

- Elle doit permettre une description complète des formes géométriques du contour ainsi que des lignes intérieures, description particulièrement utilisée lors des phases de résolution et d'exploitation des résultats

- Elle doit permettre aisément l'affectation des propriétés physiques (types de matériaux, d'efforts appliqués, de conditions limites appliqués . .etc)

- Eventuellement, elle doit permettre un remaillage aisé du domaine à partir de données de départ légèrement modifiées.

I-4 ; maillages développes et exigences

1-4-1 : en dimension 2

Les algorithmes de maillage que nous avons développés sont de deux types en dimension 2.

1 - Le domaine à mailler est pré-découpé en éléments standards. Ces éléments sont ensuite divisés en éléments finis

2 - Le domaine à mailler est défini par son contour et par des points répartis sur celui-ci. Le maillage par éléments finis est réalisé à partir de cette seule donnée, des points intérieurs étant ajoutés durant le processus par le programme lui-même.

Ces deux algorithmes sont assez différents, tant par les données qu'ils exigent que par la façon dont ils fournissent leur résultat. Pour être sûr que le résultat fourni vérifiera les exigences mentionnées plus haut, il convient de définir une structure de données la plus précise possible, mais aussi, la plus réduite possible, compte tenu de problème de place mémoire posés par les micro-ordinateurs.

La constitution d'un maillage bidimensionnel peut se décomposer en deux étapes :

1- La description du contour et des lignes essentielles du domaine

2- Le processus de maillage en lui-même, à partir des données précédentes.

Ce qui peut se schématiser comme il suit :

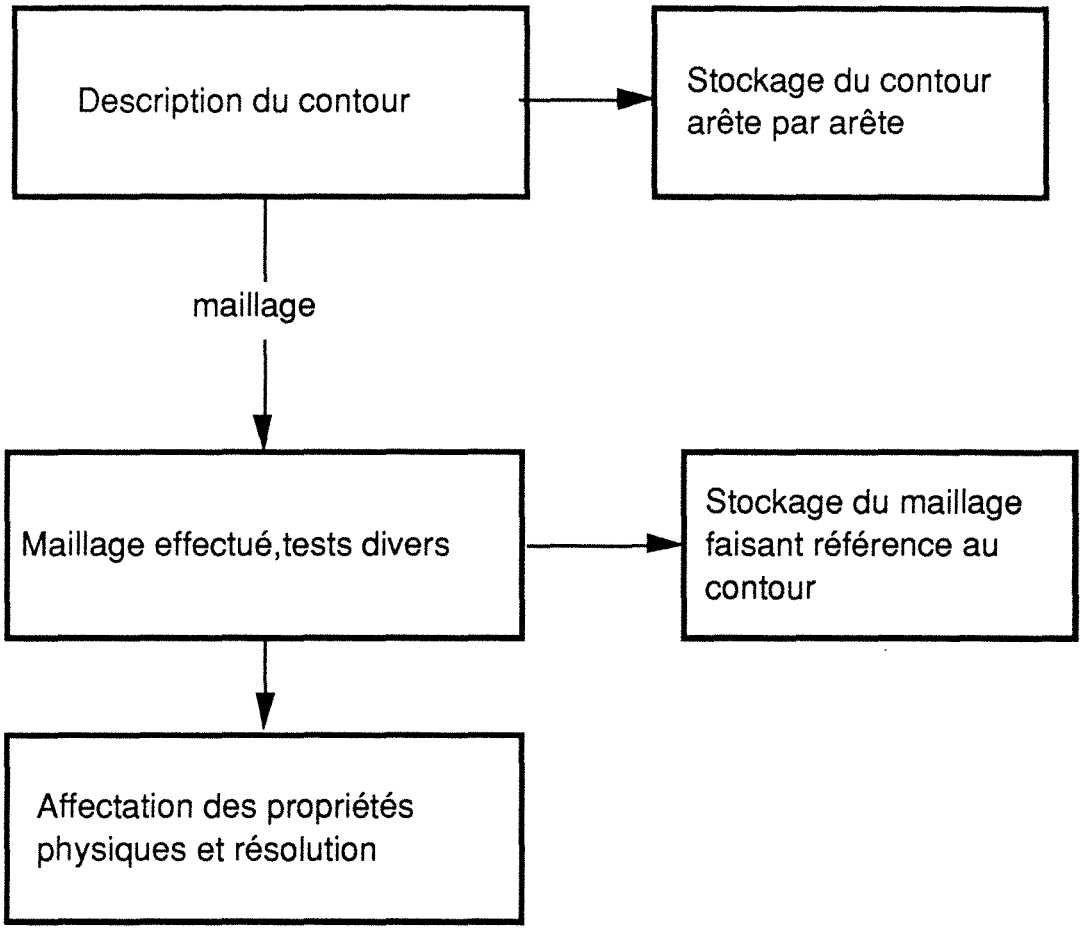
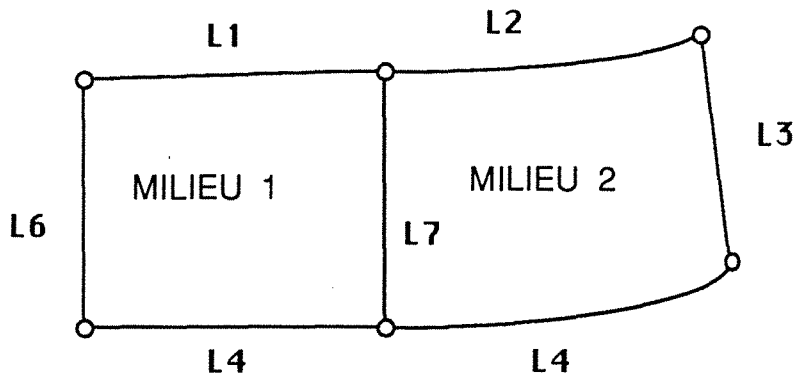


Figure I-1 : algorithme général de maillage

On le voit, la première étape est fondamentale. Le contour du domaine se compose d'une série de courbes décrivant la forme du domaine et servant à distinguer l'intérieur de l'extérieur, et d'une série de courbes intérieures permettant de préciser quelques informations supplémentaires (trous, changement de milieu, de comportement .etc).

voici un exemple:



Le contour se compose des arètes extérieures
L1-L2-L3-L4-L5-L6 et de l'arète intérieure L7

Figure I-2 : exemple de définition de contour

Cette description nécessite évidemment une étude préalable du problème. De façon plus générale, il est indispensable d'analyser le problème afin de le décomposer en problèmes plus simples. Cette analyse peut être facilitée si le domaine à mailler possède des particularités (parties se déduisant par symétrie et rotations ...). On pourra alors limiter le maillage à une partie minimale, le reste du domaine étant déduit de cette partie par les transformations géométriques appropriées.

1-4-1 : En dimension 3

Nous avons développé les algorithmes suivants :

1 -Le domaine à mailler est pré-découpé en éléments standards. Ces éléments sont ensuite divisés en éléments finis Ce programme est l'analogue de celui mis au point en dimension 2.

2 -Le domaine à mailler peut être représenté par le déplacement dans l'espace d'une surface plane (topologie cylindrique).

Dans le premier cas, nous retrouvons une structure tout à fait semblable à celle vue en dimension 2. Le second cas est un peu plus complexe. La surface de base est décrite par un maillage bidimensionnel. La donnée de cette surface et du déplacement dans l'espace suffisent à définir le contour du domaine. Dans les deux cas, cependant, nous utiliserons une représentation solide du domaine par élimination des parties cachées. Dans cette optique, nous approcherons le contour du domaine par un ensemble de facettes planes directement déduites du maillage.

Les structures de données tridimensionnelles étant particulières, nous en parlerons de façon plus approfondie au chapitre IV.

I-5 : Structure de données utilisée

1-5-1 : En dimension 2 :

Après avoir vu quelles étaient les exigences d'un maillage par éléments finis, voyons pratiquement comment sont nous avons choisi de structurer nos données. Ces informations seront stockées en mémoire centrale ou sur disque dur , selon le cas : lors des opérations de maillage, les informations géométriques (éléments, arêtes, . .etc) étant les plus utilisées, résideront en mémoire centrale, par contre lors des phases de résolution, ce sont les informations numériques (matrices, vecteurs inconnus . .etc) qui résideront en mémoire centrale, le reste étant délocalisé sur fichier disque.

- Les noeuds sont repérés par leur coordonnées stockées en mémoire centrale; aussi bien lors des phases de résolution que des phases de maillage.

- Les éléments sont mémorisés par la suite des noeuds qui le composent, pris dans un sens fixé (généralement le sens trigonométrique). Cela permet implicitement de définir la liste des arêtes orientées composant chaque élément.

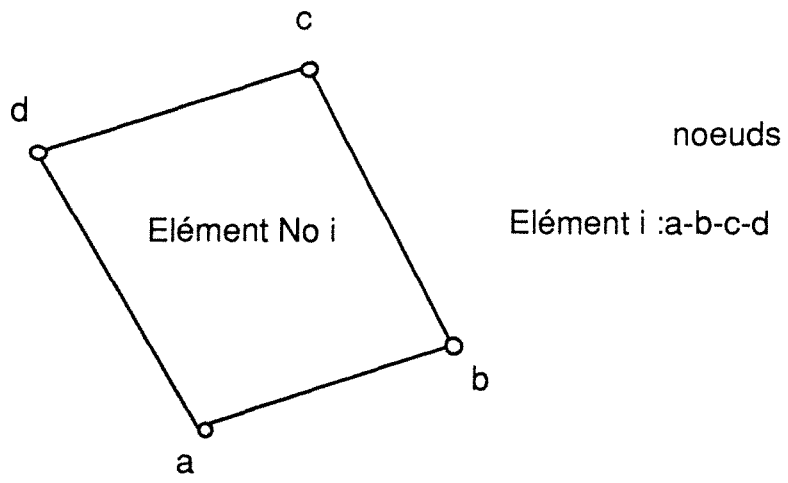
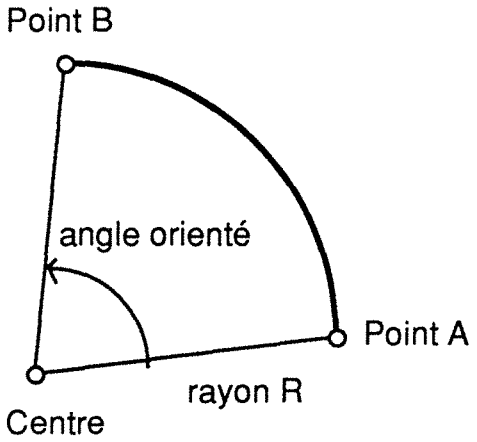


Figure I-3 : définition d'un élément par ses nœuds

A ceci ajoutent plusieurs champs utilisés lors de la résolution (type d'élément, type de matériau . . etc).

- Le contour est défini par une suite de courbes externes (départageant l'intérieur du domaine de l'extérieur) et de courbes internes (définissant, entre autres, des changement de propriétés des matériaux). L'ensemble des courbes externes est orienté de telle façon que l'on obtienne un contour fermé et sans point double.
  Les courbes utilisées se limitent aux arcs de cercle ou aux segments de droite. Leur stockage réside sur fichier disque sous la forme suivante :



courbe i: point A, point B, rayon R, X-centre, Y-centre, angle

Figure I-4 : stockage d'une arête courbe

Le sens de rotation des arêtes circulaires est pris systématiquement dans le sens trigonométrique.

- En ce qui concerne les éléments, il convient aussi de stocker les côtés frontières. Ils sont constitués par les côtés d'éléments étant assujettis à être sur une frontière donnée du domaine à mailler. Voici deux exemples élémentaires :

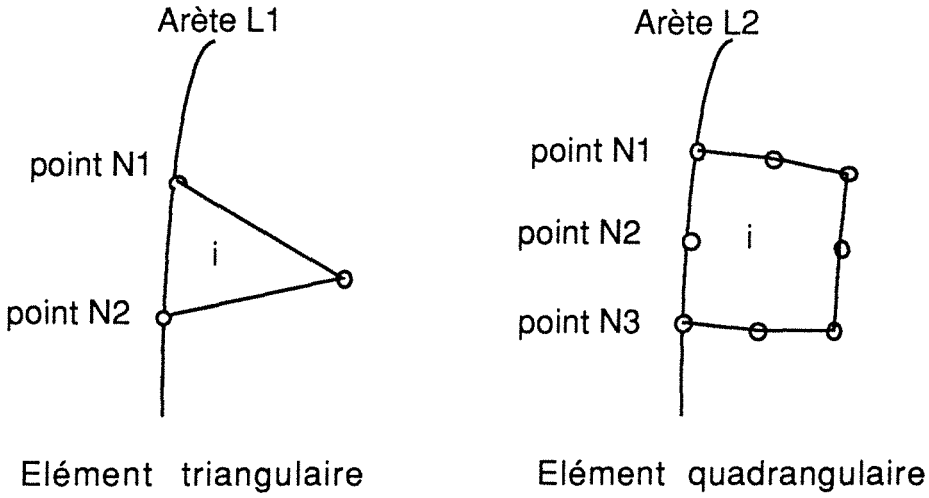


Figure I-5 : exemple de côtés frontière

Nous voyons sur l'exemple que nous pouvons avoir plusieurs noeuds par côté selon le type d'élément. Le stockage se fera de la façon suivante

1er cas : point N1 , 0 , point N2, arête de réference L1

2er cas : point N1, point N2 , point N3, arête de réference L2

Ces informations résident sur fichier disque.

Remarquons cependant que ces données ne rendent compte que de la structure géométrique de la pièce et non des conditions limites numériques appliquées (type Dirichlet, Neumann, Fourier . .). L'affectation des divers type de conditions limites sur la partie de frontière correspondante se fera durant le processus de résolution. Comme elles différent selon le type de problème étudié (mécanique, thermique . .), et le type de limite imposé, cela permet une plus grande indépendance du mailleur par rapport au solveur.

Le stockage des côtés frontière permet de savoir quel élément se situe sur quelle frontière. comme nous le verrons plus loin, beaucoup de nos méthodes font appel à des découpages d'éléments en élément plus petits. Le type de d'informations que nous mémorisons permet de diviser chaque côté frontière conformément à la courbe qu'il est censé décrire, comme nous le montre l'exemple ci-dessous.

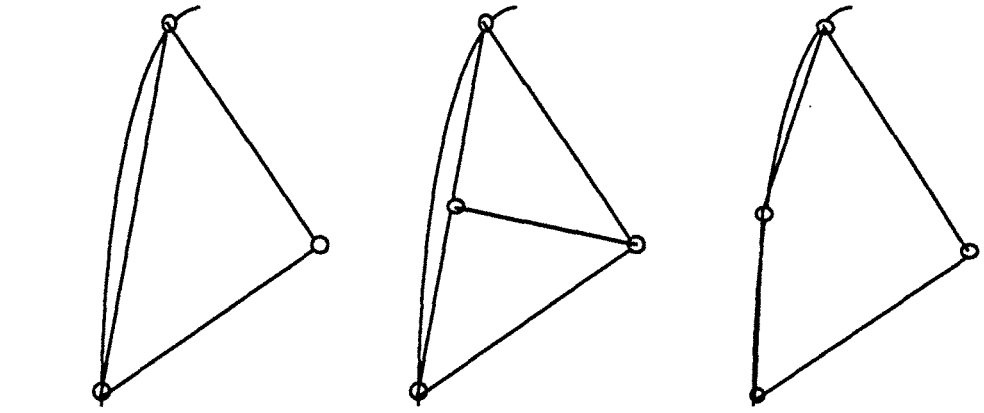


Figure I-6 : projection sur frontière courbe

Notons enfin qu'il serait possible de stocker les points se situant sur le contour (points frontières), mais ceci n'a que peu d'intérêt car :

-les points frontières se déduisent aisément des côtés frontières, l'inverse étant faux.
-un point frontière peut se situer sur plusieurs courbes frontières à la fois.

Il n'est donc pas utile de s'encombrer de ce type de données. Cependant, lors de la résolution, il sera indispensable d'avoir des renseignements sur les conditions limites numériques, ce qui donnera lieu à un stockage d'informations découplé du processus de maillage.

Enfin pour terminer, notons la réservation de plusieurs champs (liés au stockage des éléments entre autres) et utilisés pour définir des renseignements indispensables à la résolution (type de matériau,type de comportement,type de division . etc).

1-5-2 : En dimension 3 :

Comme nous l'avons fait déja remarquer, la structure de données tridimensionnelle est particulière. Nous mémoriserons les points et les éléments de la même façon qu'en dimension 2. Le maillage réalisé sera mis en valeur par une représentation solide ne tenant pas compte des parties cachées. Pour cela nous stockerons aussi les faces frontières sous la forme d'un ensemble de facettes planes. Chacune de ces facettes est décrite comme un élément plan, c'est à dire par l'ensemble de ses sommets parcourus dans un sens donné.

Chapitre II

Maillage par transformation conforme

II-1 : Introduction

L'une des idées de maillage les plus naturelles consiste à décomposer le domaine de base en sous-domaines élémentaires que l'on sait mailler. Nous présentons dans ce chapitre une application de cette méthode utilisant la représentation isoparamétrique des éléments de Lagrange plan à 8 noeuds.

Après avoir rappelé les bases théoriques de la méthode, puis développé un processus de maillage, nous présenterons tout une série d'utilitaires permettant de donner au logiciel crée sa pleine efficacité. Dans l'ensemble du travail, nous nous sommes efforcé de créer un logiciel le plus interactif possible, utilisant au maximum les possibilités du Mac Intosh, de façon à pouvoir être utilisé par le plus grand nombre.

II-2 : Principe général

Nous présentons dans ce chapitre un mailleur automatique en quadrilatères ou en triangles associés à des fonctions d'interpolation de tous degrés. Il utilise le principe de la représentation isoparamétrique par transformation conforme. Rappelons brièvement cette théorie.

Considérons un élément fini 8-nœuds bidimensionnel idéal centré sur 0 et de côté 1.

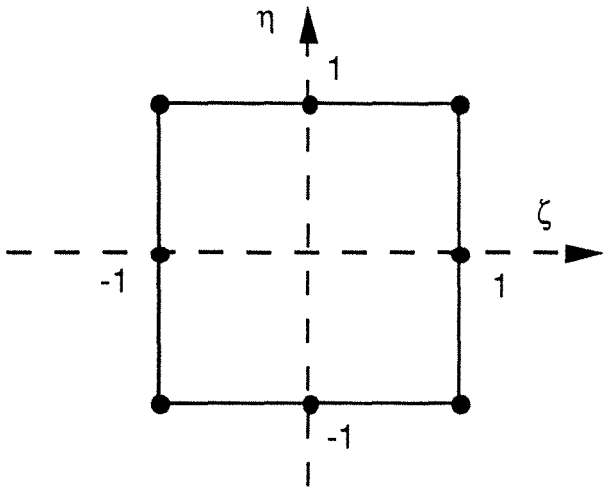


Figure II-1 : définition de l'élément de référence

A chacun de ces huit nœuds est affecté une fonction polynomiale du troisième degrés N_i (i=1,8), dite "fonction de forme", ayant la particularité de vérifier :

$$N_i(\zeta_j,\eta_j)=\delta_{ij}$$

où : δ_{ij} désigne le symbole de Kronecker.
(ζ_i,η_i) désignent les coordonnées de chaque nœud.

L'image de cet élément idéal par une certaine transformation conforme permet de définir un nouvel élément 8-nœuds appelé "élément réel" tel que l'on puisse écrire:

$$x=f(\zeta,\eta)$$
$$y=g(\zeta,\eta)$$

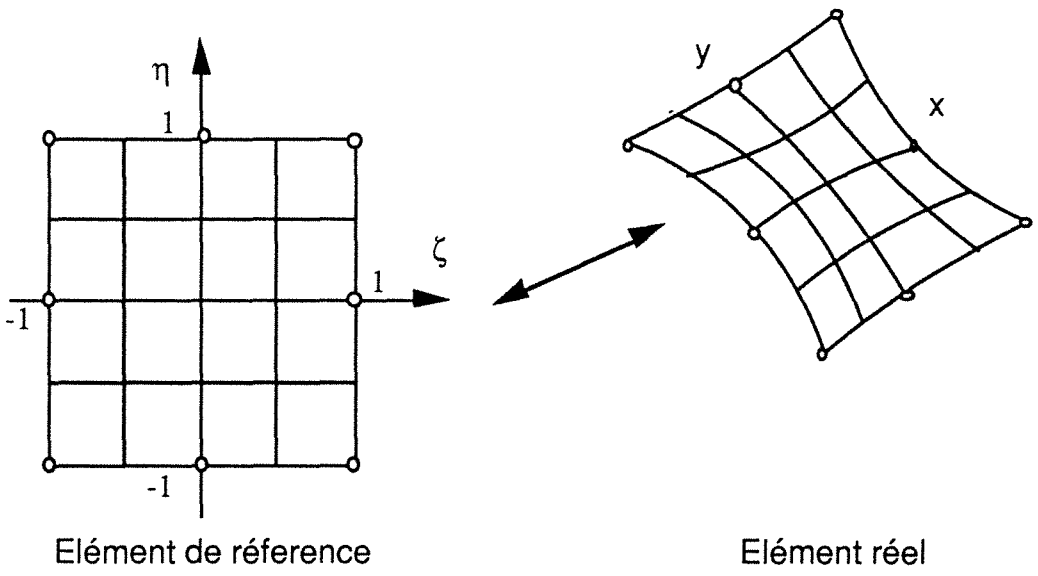


Figure II-2 : passage de l'élément de référence à l'élément réel

L'élément idéal sera désigné par la suite sous le terme d'élément de référence.

Considérons alors dans l'élément de référence un réseau orthogonal de lignes (ζ=constante et η=constante) Après transformation conforme, ce réseau est transformé en dans l'élément réel en un réseau toujours orthogonal conformément à la figure II-2 ci-dessus.

Ces courbes sont appelées "courbes isoparamétriques"

On voit donc qu'il est possible, à partir de n'importe quel élément 8-nœuds dont on se donne une densité de points répartis en vis à vis sur les frontières, de générer par l'intermédiaire de l'élément de référence, des points internes situés sur un réseau de courbes orthogonales entre elles. Ceci nous permet donc d'obtenir des quadrilatères élémentaires dont l'intérêt en vue d'un processus de maillage est évident.

Si l'on sait décomposer le domaine à mailler en un ou plusieurs éléments 8-nœuds, cette méthode nous permettra d'obtenir un maillage correct au sens des éléments finis.

De façon générale, on utilise les fonctions de forme pour représenter toute variation d'une variable sur l'élément réel en fonction de ses valeurs aux nœuds. Cette relation s'écrit pour une inconnue u quelconque

$$u = \sum_{i=1}^{n} N_i(\xi,\eta).u_i$$

et en particulier :

$$x = \sum_{i=1}^{n} N_i(\xi,\eta).x_i \quad \text{et} \quad y = \sum_{i=1}^{n} N_i(\xi,\eta).y_i$$

Où : _ N_i est la fonction de forme exprimée dans le repère de référence affectée au nœud numéro i

_ u_i est la valeur de la variable u au nœud numéro i

_ (ζ,η) sont les coordonnées du point considéré dans le repère de référence

_ n est le nombre de nœuds de l'élément.

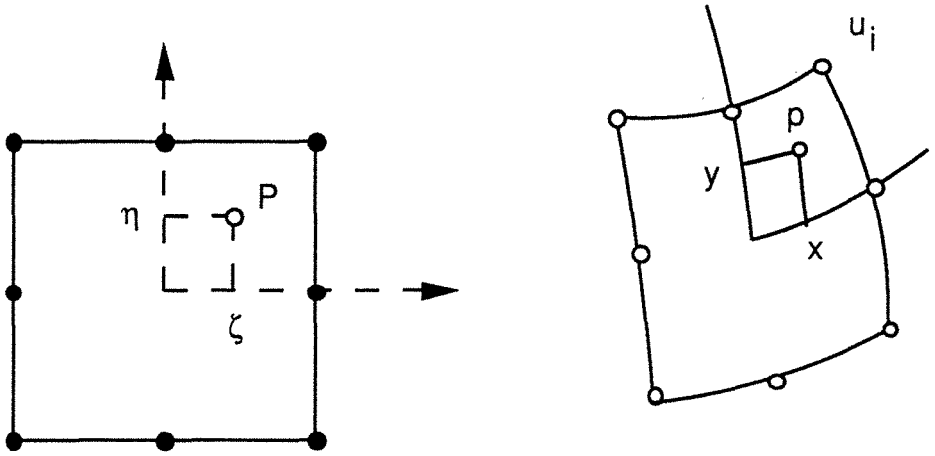


Figure II-3 : approximation nodale de la variable u

Pour qu'une transformation conforme soit acceptable, il faut qu'elle provoque une description "raisonnable" de l'élément réel. Si cela n'est pas le cas, il n'y a plus unicité de la définition, le procédé de transformation devient inutilisable

S'agissant de transformation conforme, nous aurons unicité de la transformation sur tout domaine où le Jacobien de la transformation (ζ,η) -> (x,y) est non nul. Cela implique en particulier que le signe de cette transformation garde un signe constant [EL 1].

Dans les cas simples, certaines règles faciles à mettre [EL 1] en oeuvre assurent l'unicité de la transformation. Pour un quadrilatère 4-nœuds, par exemple, chaque angle doit être inférieur à 180°, conformément à la figure II-4 :

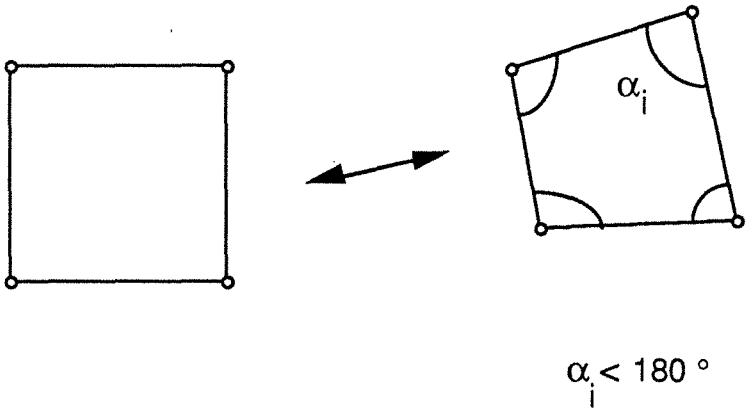


Figure II-4 : condition de validité d'un quadrilatère 4 nœuds

De même, pour un élément 8-noeuds (élément que nous utiliserons), il est impératif d'imposer une transformation vérifiant :

- Les angles α_i définis sur la figure II-5 doivent être inférieurs à 180 °
- Le nœud central sur chaque côté doit être compris dans le tiers central de ce côté, comme le montre la figure II-5

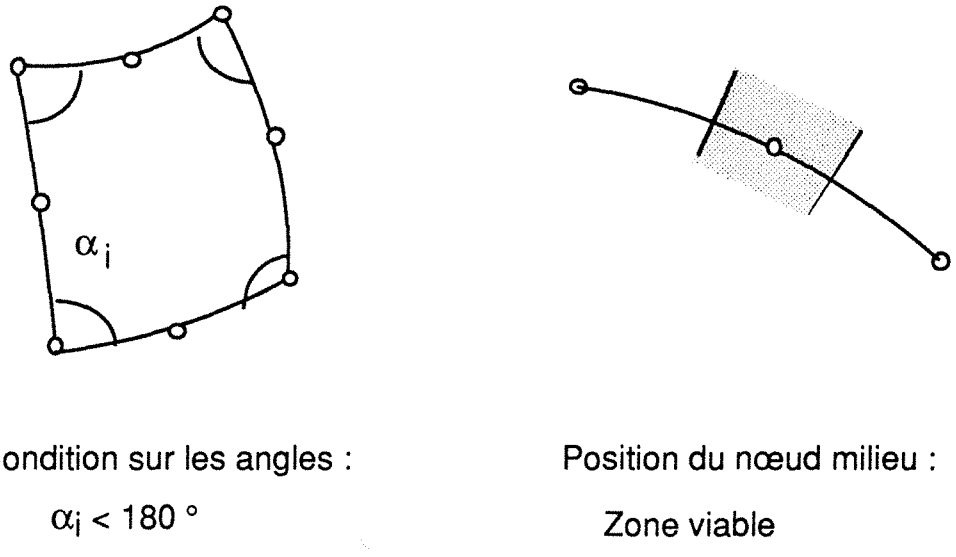
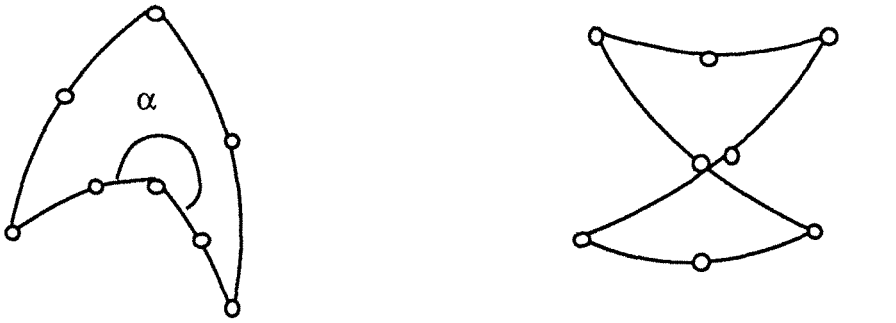


Figure II-5 : validité d'un élément 8 nœuds

- Lorsque l'on parcourt le contour de l'élément, l'intérieur du domaine doit être toujours du même côté (généralement à gauche lors d'un parcours dans le sens trigonométrique).

Il en résulte, par exemple,que les deux éléments suivants sont inutilisables :



a) $\alpha > 180°$

b) Intérieur du domaine changeant de côté le long du contour

Figure II-6 : exemple d'éléments incorrects

Ces simples règles suffisent, dans le cas de ces deux éléments à assurer l'existence et l'unicité de la transformation. Pour des élément associés à des fonctions de forme cubiques ou plus, de telles régles sont généralement inapplicables et il est nécessaire de vérifier numériquement le signe du Jacobien [EL 1].

Le maillage obtenu doit vérifier un certain nombre de règles :

- Il doit être conforme à la définition d'un maillage par éléments finis (voir chapitre I).
- Il doit assurer la continuité de l'information entre éléments.

En ce qui concerne le premier point, le fait d'utiliser toujours le même type d'élément (8-nœuds) assure la continuité des coordonnées x et y. Il n'y a donc pas de "trous" dans le maillage.

Enfin nous assurons la continuité de toute variable u sur le domaine à mailler par le théorème suivant [EL 1]:

- Si les fonctions de forme Ni sont telles que, dans l'élément de référence, la continuité de la variable u est préservée, alors, la continuité de u sera vérifiée sur les éléments réels.

Cet ensemble de résultats permet à la fois de valider le processus de maillage tout en en fixant les limites

Notons que cette notion de courbe isoparamétrique se généralise à tous les types d'éléments finis classiques. Il est par ailleurs possible de définir sur des domaines de formes plus générales des procédés similaires par interpolations généralisées [EL 9].

Nous avons choisi l'élément isoparamétrique 8-nœuds pour sa facilité de définition, sa souplesse d'emploi et sa relative performance. Les autres formulations aboutissant soit à de moindres performances, soit à des calculs plus complexes.

Notons enfin que les points situés sur deux côtes opposés de chaque élément 8-nœud doivent être parfaitement en vis à vis. Par contre, il peuvent très bien ne pas être équidistants, ce qui permet d'affiner le maillage selon une ou plusieurs directions données.

II-3 : Méthode de maillage par blocs

Bien que la méthode de maillage développée soit générale, nous avons choisi de nous limiter le type d'éléments par ce processus aux triangles 3 et 6 nœuds ainsi qu'aux quadrilatères 4 , 8 et 9 noeuds. Il serait possible de mailler avec des éléments plus riches mais on y perdrait des informations (configuration sub-paramétrique), nous avons donc exclu cette possibilité.

Donc, la première étape va consister à décomposer le domaine à mailler en éléments 8-nœuds grossiers (si cela est possible), éléments que nous appelerons "blocs" dans la suite de l'exposé.

Dans une deuxième étape, chaque bloc est maillé séparément en construisant les points intérieurs couche par couche et en les reliant entre eux, construisant ainsi des "mailles de base" de forme quadrilatèrale ..

L'orthogonalité des deux réseaux d'isoparamétriques permet d'obtenir des quadrilatères bien équilibrés et de bonne qualité, surtout si les points sur les frontières sont équidistants.

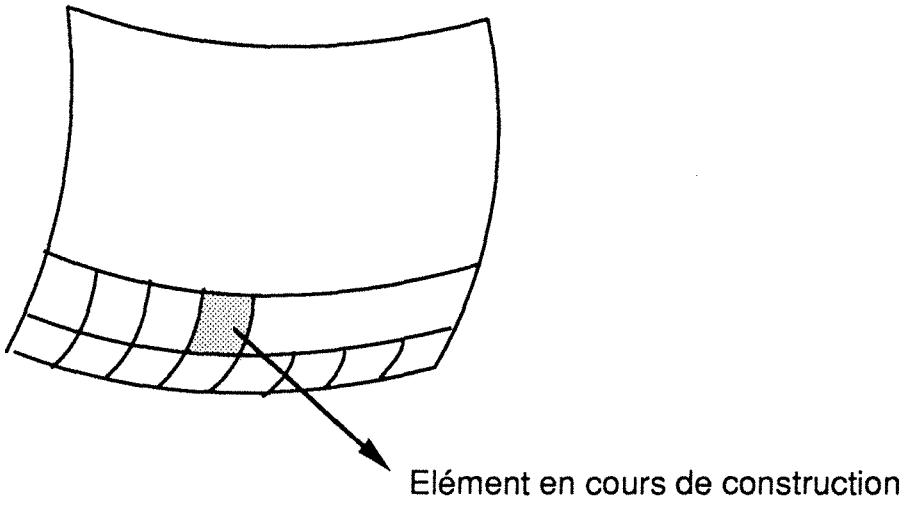


Figure II-7 : schéma de construction du maillage

Les blocs sont maillés indépendamment les uns des autres et les mailles de bases quadrilatérales sont stockées.

Le processus crée des points identiques appartenant à des blocs différents (donc des points de numéros différents ayant des coordonnées identiques). Une fois le maillage de chaque bloc réalisé, on fusionne ces points, et la correction correspondante est introduite dans les éléments stockés. Enfin, dernière étape, selon le cas (éléments triangulaires ou 9-nœuds) des points supplémentaires sont ajoutés et les éléments sont divisés de façon à obtenir le résultat souhaité. Cette division se fait selon la plus courte diagonale pour améliorer la qualité du maillage crée.

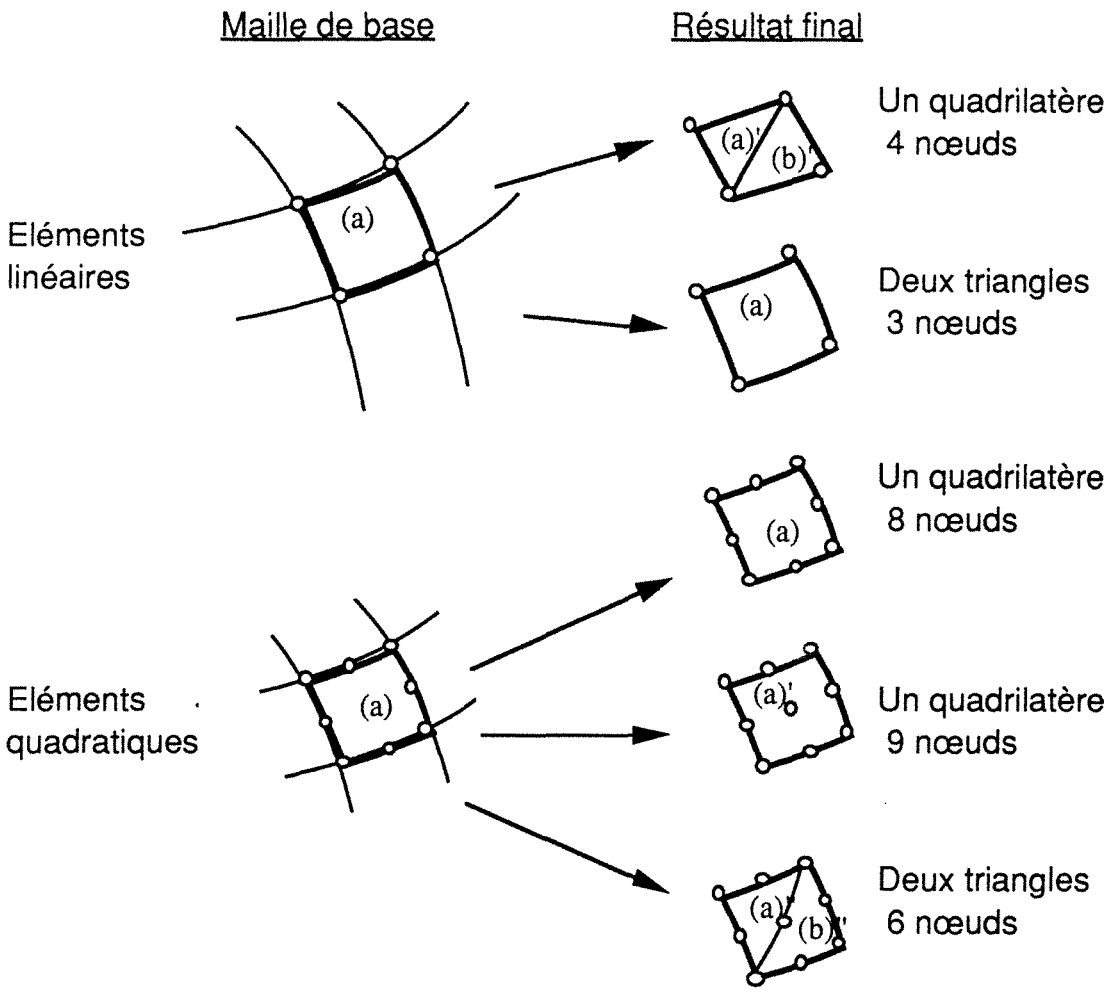


Figure II-8 : principe de construction des éléments

On peut résumer notre méthodologie par l'algorithme suivant :

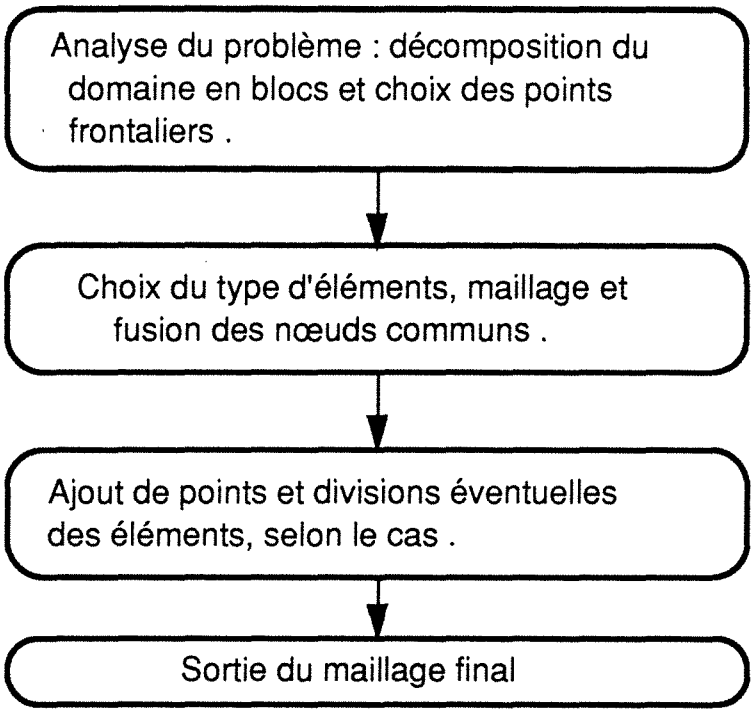


Figure II-9 : algorithme général de maillage

II-4 : Gestion et stockage des données de base

La première étape du travail consiste à décomposer le domaine à mailler en blocs. Cette étape n'est pas difficile, mais elle est parfois pénible. Nous avons donc mis au point un logiciel interactif travaillant uniquement par l'intermédiaire de menus déroulants et de saisies directes à l'écran. Ce travail a été effectué avec l'aide de la Toolbox Mac Intosh,bibliothèque regroupant toutes les procédures graphiques et informatiques de gestion du système d'exploitation Mac Intosh.

Les données nécessaires à la définition des blocs sont stockées en mémoire centrale, soit à partir d'un fichier précédemment sauvegardé, soit en définissant les blocs dans le cadre d'un nouveau problème. De nouveaux blocs peuvent être crées , soit par saisie directe, soit en utilisant des transformées géométriques simples (déplacement ou duplication par symétrie, translation, etc). Ce travail est facilité par le calcul automatique de formes simples (rectangles, ovales etc).

La gestion du contour du domaine se fait de façon automatique. La description des blocs par leur contour orienté définit de façon implicite quatre arêtes circulaires ou droites, définie chacune par trois points.

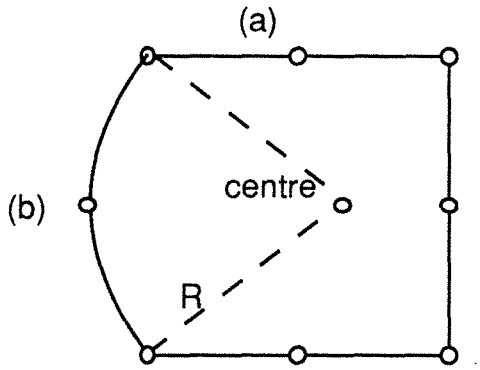


Figure II-10 : définition des arêtes d'un bloc

Si ces trois points sont alignés, on considère l'arête comme droite, sinon, on construit un arc de cercle circonscrit à ces trois points d'où l'on déduit le centre, le rayon et l'arc. Dans l'exemple ci-dessus (figure II-12), l'arête (a) du contour est droite, l'arête (b) est circulaire. Ces arête sont stockées conformément à la structure de données vue au chapitre I.

Lors de l'apparition de blocs nouveaux, les arêtes frontières sont testées de façon à éliminer celles qui sont devenues internes (arête commune intégralement à deux blocs). Ceci permet une gestion automatique du contour sans que l'utilisateur n'ait à intervenir.

Lorsque tous les blocs nécessaires ont été donnés, l'utilisateur doit choisir la densité d'éléments sur chaque direction. Il a le choix entre une division équidistante ou une division semi-logarithmique centrée sur un point, ce qui permet de concentrer le maillage dans une zone données.

L'ensemble des blocs constituant le domaine est stocké sur un fichier disque (ce qui en permet la ré-utilisation en le modifiant ou non). La structure de données est compatible avec celle décrite au chapitre I , elle se compose essentiellement de :

- l'ensemble de points nécessaires à la définition des blocs.

- la description de chaque bloc par l'ensemble de ses points décrivant le contour du bloc dans le sens trigonométrique.

- L'ensemble des arête frontières décrivant le contour du domaine

A toutes ces procédures de calcul, s'ajoutent un grand nombre d'utilitaires graphiques (dessin,zoom,numérotation, etc) visualisant automatiquement le travail crée. De plus, un grand nombre de contrôles automatiques ont été mis en place pour détecter et signaler toute entrée aberrante.

A tous niveaux, il est possible de détruire tout ou partie du travail crée tout en conservant le reste (si tenté qu'il ait un reste). On peut ainsi recommencer le travail en modifiant certaines informations (les densités d'élément par exemple). Dans tous les cas, la structure de données (et particulièrement le contour du domaine) est modifié en conséquence afin d'être compatible avec le problème posé.

Voici l'exemple d'une plaque carrée percée d'un trou rond. Le maillage donné se décompose en quatre blocs et 20 nœuds de référence (figure II-11)

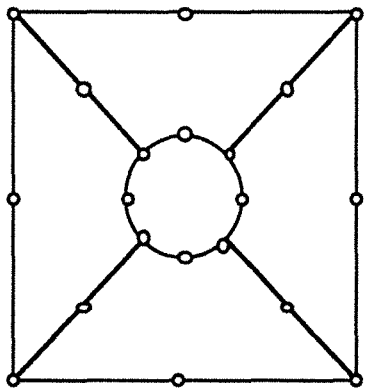


Figure II-11 : Décomposition du domaine en blocs

Le programme gère automatiquement le contour suivant:

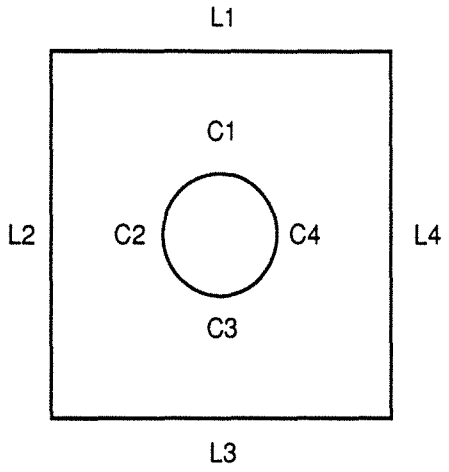


Figure II-12 : Le contour correspondant

Il se compose de 4 arêtes droites (L1 à L4) et de quatre arêtes circulaires (C1 à C4) qui permettrons de définir le domaine quel que soit le maillage souhaité.

Une fois que l'utilisateur a fini de définir son domaine, il choisit les densités d'éléments sur les bords ainsi que le type d'éléments choisis.

Vient alors la phase de maillage proprement dite. Lors de cette phase, les variables utilisées pour stocker les blocs sont détruites, nous ne conservons que les données concernant le contour. Le résultat du maillage après opérations complémentaires (fusions des nœuds communs et divisions éventuelles) est stocké en mémoire centrale. Il comprend:

- les nœuds
- les éléments
- les côtés frontières

Ces résultats sont conservés conformément à la structure de données vue au chapitre I.

II-5 : Utilitaires locaux et globaux de modification de maillage

Si les procédures mentionnés dans les paragraphes précédent sont efficaces et conviviaux, ils ne permettent cependant pas de couvrir tous les cas auquel on peut être confronté.

Il nous a donc paru indispensable d'adjoindre toute une série d'utilitaires de travail "a posteriori", c'est à dire travaillant sur le maillage fini et non sur les données de base, et ce pour plusieurs raisons :

- il n'est pas possible de traiter toutes les formes possibles. L'introduction de nouveaux types de formes (par exemple l'introduction de bloc triangulaires) aurait compliqué le logiciel.

- De plus, l'introduction de formes nouvelles se serait soldé par des performances de maillage moins bonnes (qualité des éléments) et/ou l'élimination de certains types d'éléments.

Nous avons donc décidé de travailler sur le maillage fini. La plupart de ces utilitaires concernent des modifications ou des suppressions locales d'entités (points ou éléments). Ces dispositions permettent particulièrement de traiter rapidement des problèmes compliqués qui ne sont que des perturbations de problèmes simples.

Comme précédemment, nous avons voulu développer des fonctions le plus interactives possibles par l'utilisation de menus déroulants et de saisies directes à l'écran. Mais surtout nous désirons que ces fonctions soient le plus "sûres" possible. Par "sûr", nous entendons le fait que toute modification du maillage doit créer un résultat compatible avec la notion d'éléments finis sans que l'utilisateur n'ait à intervenir.

Nous devons donc faire continuellement appel à des modifications de la structure de données (points, éléments,côtés frontières, etc). Ces opérations sont plus complexes et donc plus longues que celles nécessitées par la gestion des blocs.

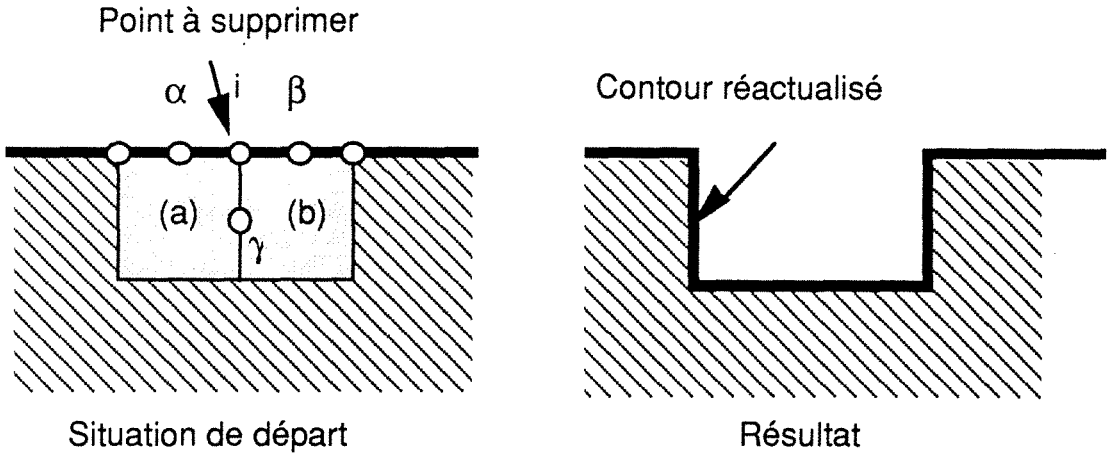
Aucun logiciel n'étant sûr à 100%, de nombreux utilitaires graphiques permettent de tester la qualité des résultats crées en permettant de les visualiser à tous instants.

Parmi les nombreuses fonctions crées, citons

II-5-1 : Destruction de points ou d'éléments

Ces fonctions éliminent toutes les entités saisies à l'écran, ainsi que toutes celles qui n'ont plus de sens après la destruction des premières. Ainsi, par exemple, la destruction d'un point entraîne celle des éléments contenant ce point, ce qui peut conduire à éliminer d'autres points encore. Le programme corrige automatiquement les données par suppression et décalage. De plus, il corrige également le maillage graphiquement

La figure II-13 donne un exemple de suppression de point :



La suppression du point i entraine la dispariton des éléments (a) et (b) , puis des points α, β, γ , et une réactualisation du contour .

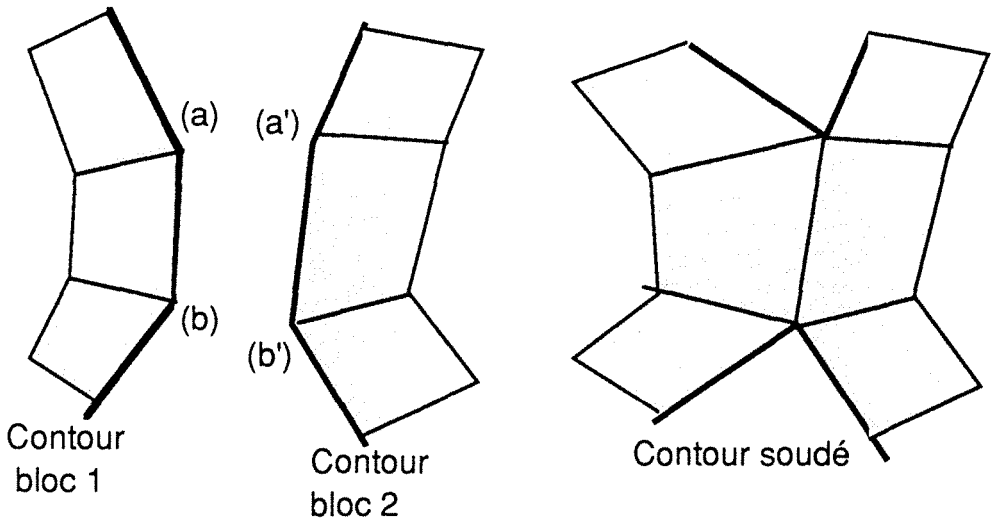
Figure II-13 : suppression de points

II-5-2 : Mouvements et collages de points

Cet utilitaire présente deux fonctions :

- La première est déformer localement le maillage, de façon manuelle par l'intermédiaire de la souris. La direction et l'amplitude du déplacement sont intégralement définis par le mouvement de la souris, donc par l'utilisateur.
- La seconde effectue les mêmes opérations que la première, mais en plus elle permet de souder ensemble deux points qui, par la suite, ne feront plus qu'un.

La gestion graphique de ces utilitaires déforme graphiquement et instantanément les éléments liés au point bouge comme s'ils étaient élastiques, permettant ainsi une vision et un contrôle des résultats. Comme précédemment, les corrections correspondantes se font automatiquement sur l'ensemble du maillage (points, éléments, côtés frontières etc)

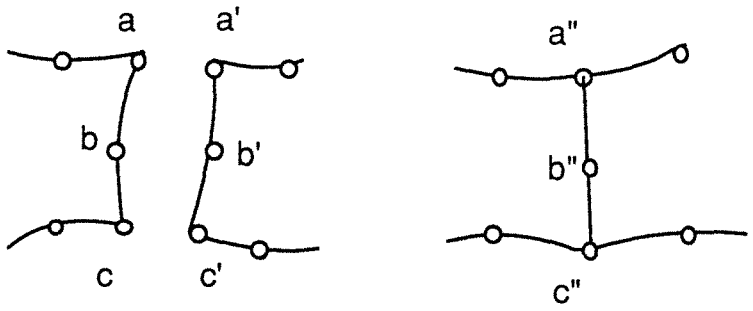


Dans l'exemple ci-dessus, on a du définir deux blocs séparés .Une fois le maillage réalisé, on soude les points (a) et (b) aux points (a') et (b') respectivement . Deux côtés frontières ont disparus dans l'opération et le contour à été réactualisé .

Figure II-14 : mouvement/collage de points

Il est à noter que si ces opérations sont assez simples pour les éléments linéaires, elles sont par contre plus complexes pour les autres types d'éléments (quadrilatères 8 et 9 nœuds, triangle 6 nœuds), notamment pour les nœuds milieux (rappelons en effet que pour être valides, de tels éléments doivent avoir leur nœuds milieux compris dans le tiers central du côté). Nous imposons donc les conditions suivantes :

- Les nœuds milieux ne peuvent pas être déplacés. Il sont par contre automatiquement soudés dès que les deux nœuds sommets composant l'arête sont collés, ce qui rend les éléments traités compatibles avec leur critères de définition, comme le montre le schéma suivant (figure II-15):



Le collage des points a et c aux points a' et c'
respectivement entraine le collage automatique de b à b'

Figure II-15

- Si un nœud sommet est déplacé, les nœuds milieux des arête correspondantes sont réactualisés en utilisant une configuration sub-paramétrique, comme le montre le dessin ci-dessous .(figure II-16)

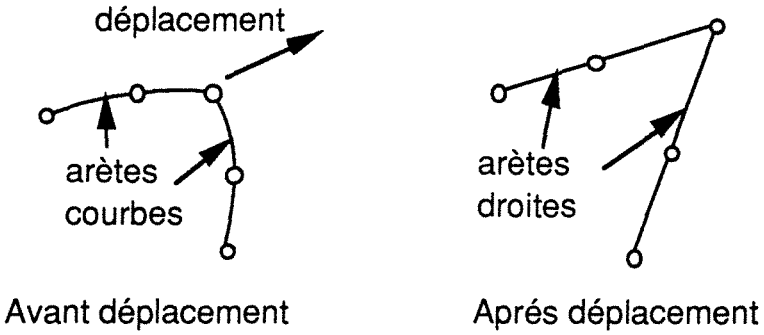


Figure II-16 : cas des éléments quadratiques

On remarquera par ailleurs que ces opérations peuvent nuire à la qualité du maillage en produisant des éléments très distordus. Un contrôle et une modification de la qualité du maillage est cependant quasiment impossible à ce niveau pour des problèmes de temps de calcul et de quantité d'informations nécessaire.

II-5-3 : Projection sur une frontière droite ou courbe

Cet utilitaire permet de projeter orthogonalement une série de points sur une courbe donnée (droite ou circulaire). Cette courbe va servir à définir une nouvelle frontière qui sera ajoutée à celles du contour existant déjà.

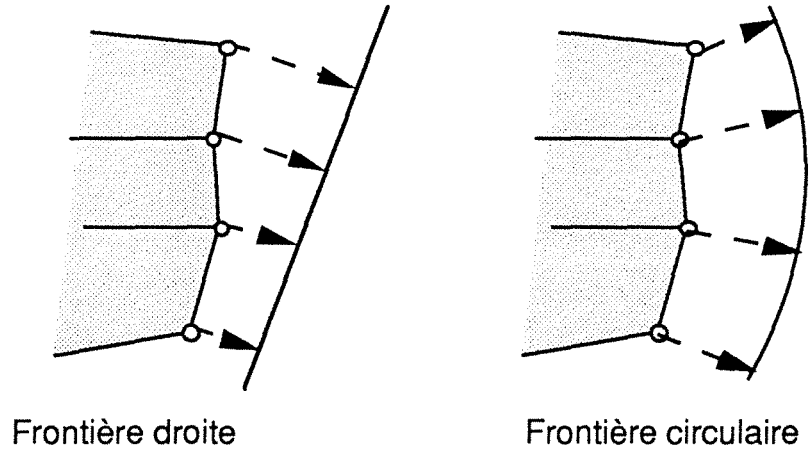


Figure II-17 : projection sur une courbe

Comme précédemment, dans le cas d'éléments isoparamétriques, les points milieux sont réactualisés en utilisant la version sub-paramétrique de l'élément, comme indiqué ci-dessous .(figure II-18)

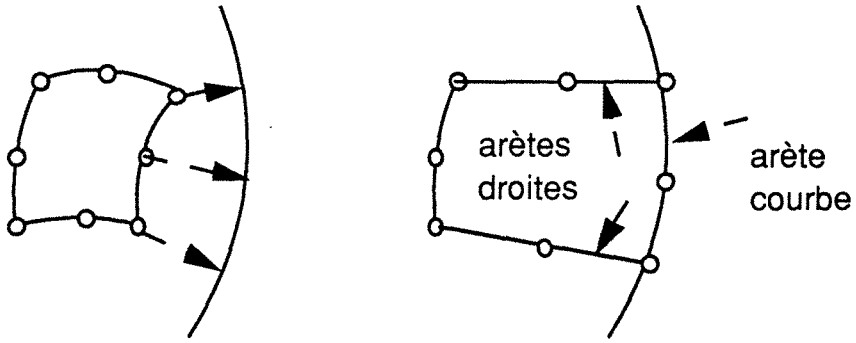


Figure II-18 : traitement des éléments quadratiques

II-5-4 : Utilitaires géométrique de reproduction de maillage

Ces utilitaires sont tout à fait semblables à ceux utilisés pour l'entrée et la définition des blocs. Il ont pour but de faciliter la gestion du maillage en reproduisant automatiquement des motifs qui se répèteraient dans le domaine; ils utilisent donc essentiellement des fonctions de symétries, de rotations et de translations.

Mais, par reproduction automatique, nous entendons une redéfinition fiable du maillage obéissant à la structure de données imposée au stockage des informations. Il convient donc de repérer les entités qui disparaissent ou qui doivent être redéfinies, en particulier:

- Les points sur les axes de transformation (invariants)
- Les arête frontières devenant internes
- Les côtés frontières se retrouvant à l'intérieur du domaine

Ces opérations sont faites pas à pas lors de la transformation de façon à reproduire un maillage total compatible et cohérent. Le croquis suivant nous donne un exemple de fonctionnement.

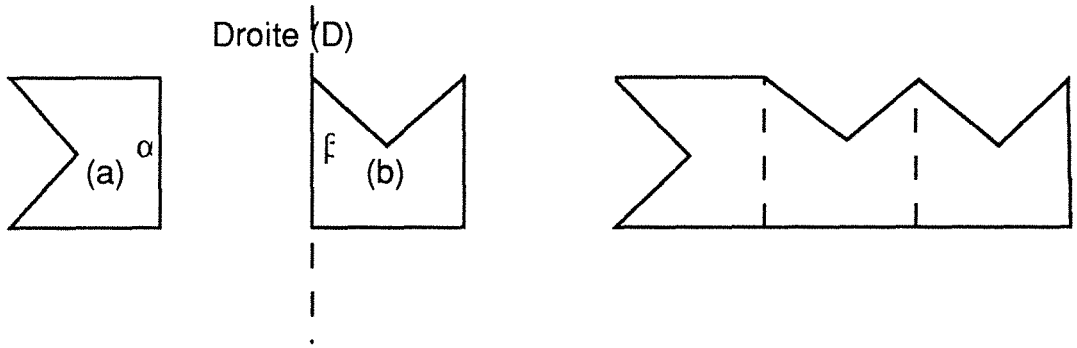


Figure II-19 : transformation géométrique

Il convient cependant d'être attentif à la partie de maillage que l'on va transformer. On ne peut en effet travailler que sur un ensemble de données cohérentes comprenant les points, les éléments, les arête et les côtés frontières, c'est à dire l'ensemble du maillage en cours. L'entrée dans ce genre d'utilitaires définit donc "un motif de base" qui est initialisé à l'ensemble du maillage. Après transformation, ce motif peut être réactualisé à l'ensemble du maillage obtenu ou non, selon le souhait de l'utilisateur; ce que nous pourrons résumer par le schéma suivant :

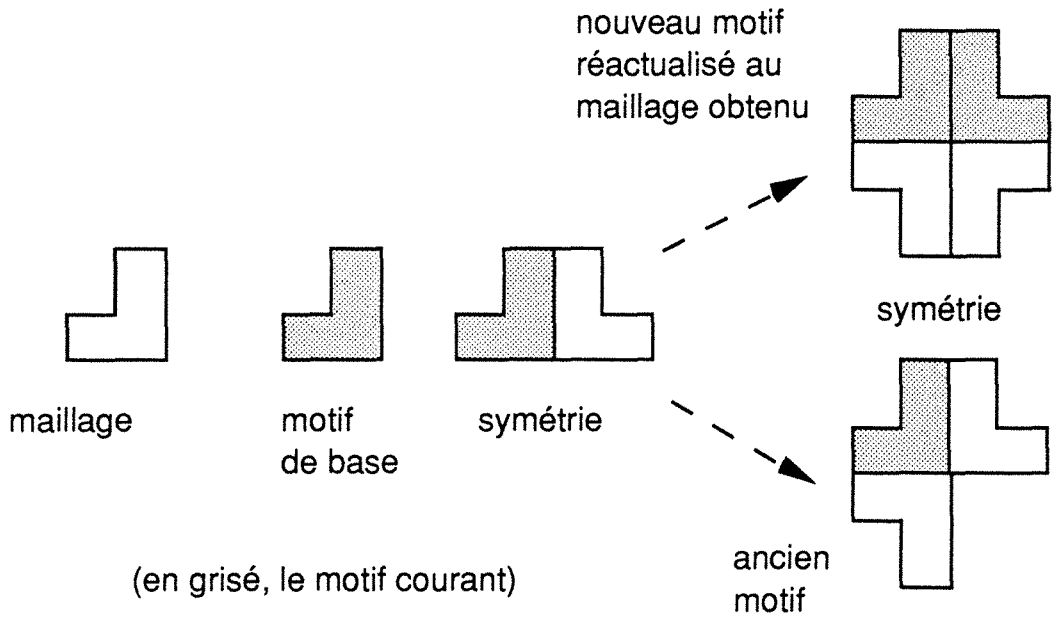


Figure II-20 : définition de motifs

D'un point de vue pratique, cet utilitaire travaille en trois étapes :

- 1 : Calculer l'image de chaque point par la transformation et tester si cette image n'existe pas déjà.
- 2 : A partir du résultat de l'étape précédente, construire les nouveaux éléments (éventuellement inversés dans le cas des symétries) par leurs connections.
- 3 : Repérer et éliminer arête et côtés frontières devenus internes au domaine.

II-5-5 : Inversion de diagonale

L'utilisation de cet utilitaire est limité aux éléments triangulaires. Cette fonction permet d'inverser la diagonale séparant deux éléments adjacents (s'ils sont compatibles) conformément à la figure II-21 ci-dessous.

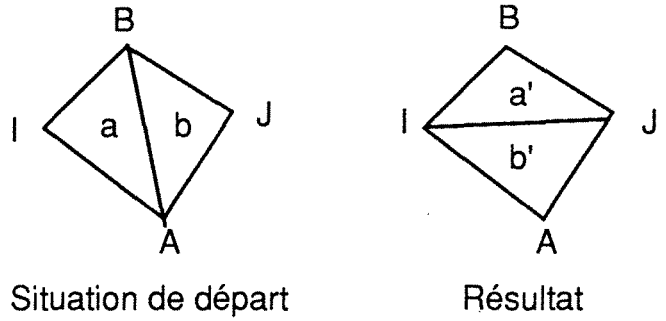


Figure II-21 : inversion de diagonales

Cette opération ne crée aucune entité. Cependant, pour que les éléments soient compatibles, il faut (avec les notation de la figure II-22 ci-dessous) que l'arête [I,J] coupe la droite (A,B) dans le segment [A,B].

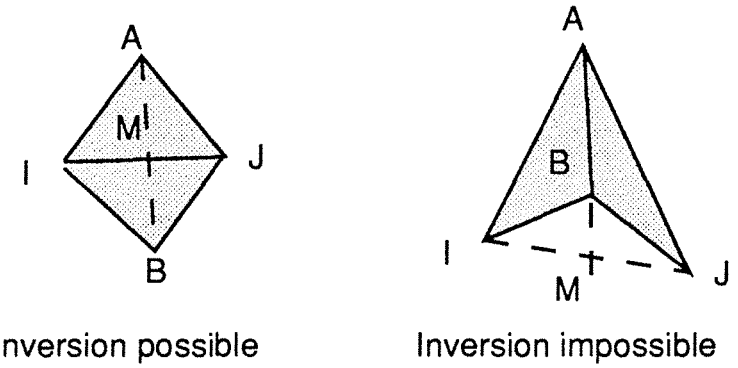


Figure II-22 : exemples de cas pratiques

Cela permet entre autres :

- De symétriser un maillage manuellement
- D'améliorer la qualité d'un maillage
- De rendre un maillage compatible avec certaines conditions limites (conditions de flux imposé par exemple)

II-5-6 : Conclusions sur ces utilitaires

Voici donc l'ensemble des utilitaires disponibles dans le programme.Ils sont gérés par menus déroulants utilisant une saisie directe à l'écran. Ils permettent de "découper" ou de modifier à volonté le maillage tout en gardant une structure de données solide et compatible. Ceci n'est acquis, comme nous venons de le voir, qu'au prix d'une constante _ et pénible _ restructuration des informations. Globalement, le temps de calcul se décompose comme suit:

- 25% pour la saisie à l'écran.
- 25% pour la gestion graphique des modifications.
- 50% pour la coûteuse restructuration des données.

Nous allons donner ci dessous l'exemple de la complexité d'un des utilitaires mis en service dans le logiciel. Il s'agit de la fonction supprimant des éléments par saisie directe à l'écran :

-a) :Saisie à l'écran des éléments .
-b) :Test sur tous les éléments pour en déduire les points à détruire .
-c) :Destruction effective des éléments et décalage dans la liste des éléments .
-d) :Destruction effective des points et décalage dans la liste des points, puis décalage en conséquence dans les connections d'éléments .
-e) :Destruction graphique des points et des éléments à éliminer .
-f) :Test et élimination des côtés frontières appartenant aux éléments détruits, avec décalage du fichier correspondant .
-g) :Rajout des nouveaux côtés frontières apparus lors de la destruction des éléments .

Tableau II-1 : organigramme de destruction de points

II-6 : Algorithme général du programme

Nous présentons le schéma complet du déroulement du programme sous la forme suivante :

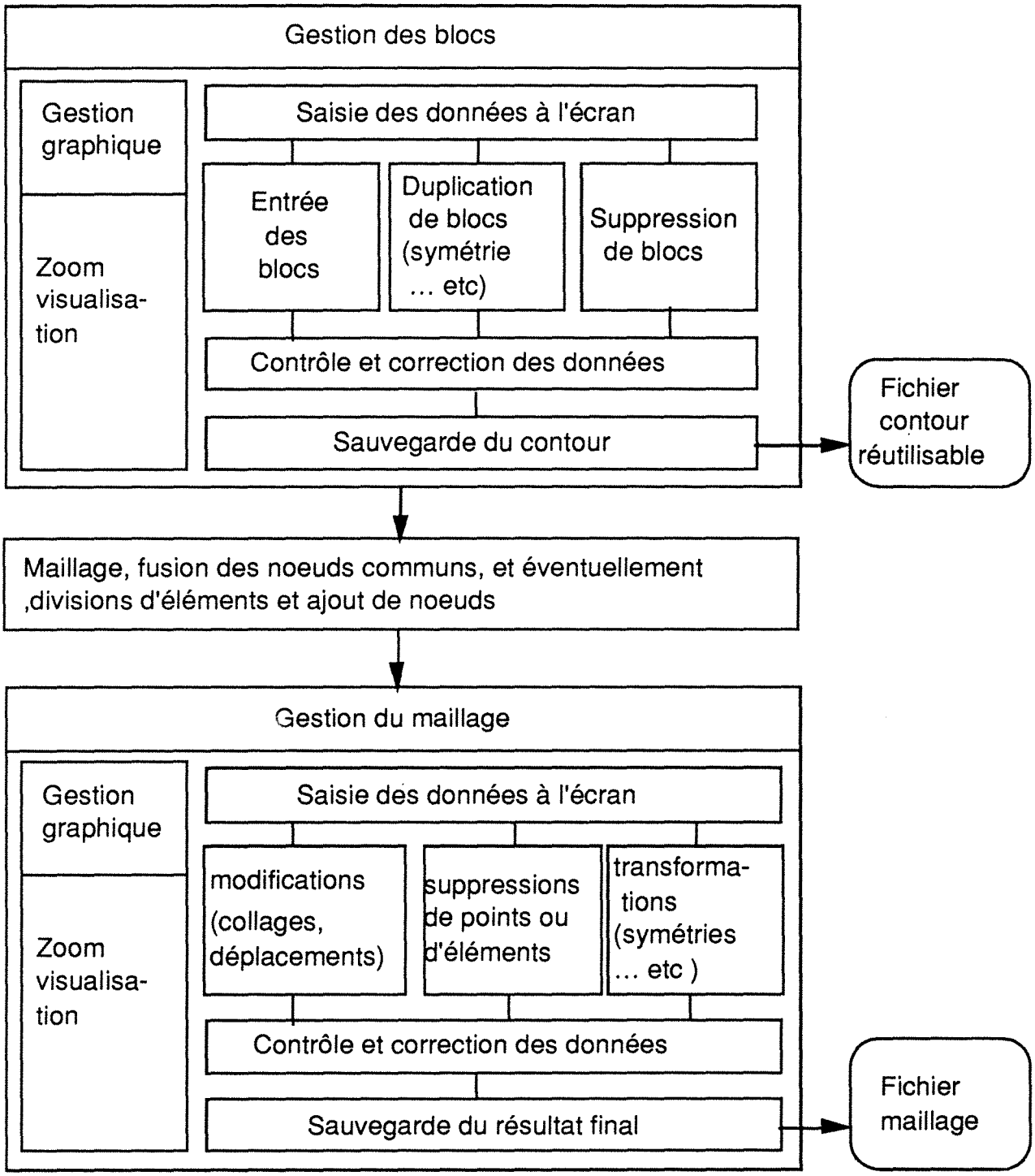


Figure II-23: algorithme général du programme

II-7 : Exemples

Nous allons dans ce paragraphe présenter quelques réalisations illustrant les possibilités du logiciel

II-7-1 : Utilisation de destructions et de modifications d'entités

Le premier exemple, construit à partir de deux bloc montre les possibilités de post-traitement d'un maillage et le résultat en découlant quant à la structure des données :

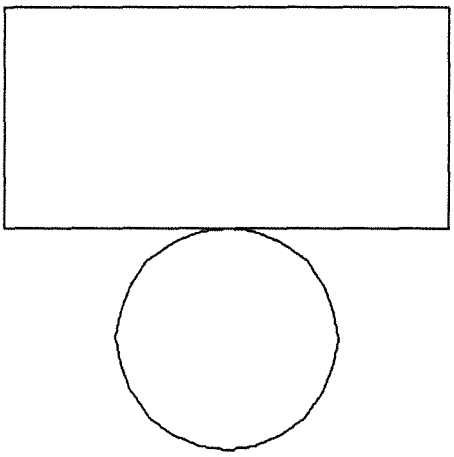


Figure II-24
Le contour des deux blocs

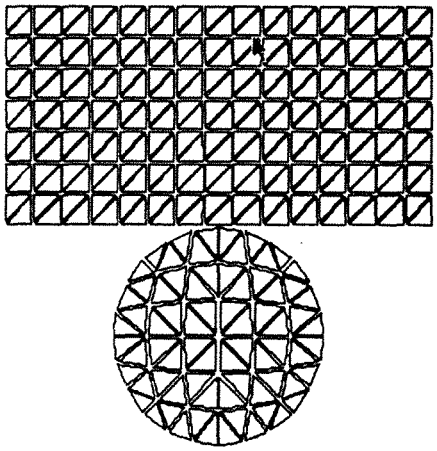


Figure II-25
Le maillage résultant

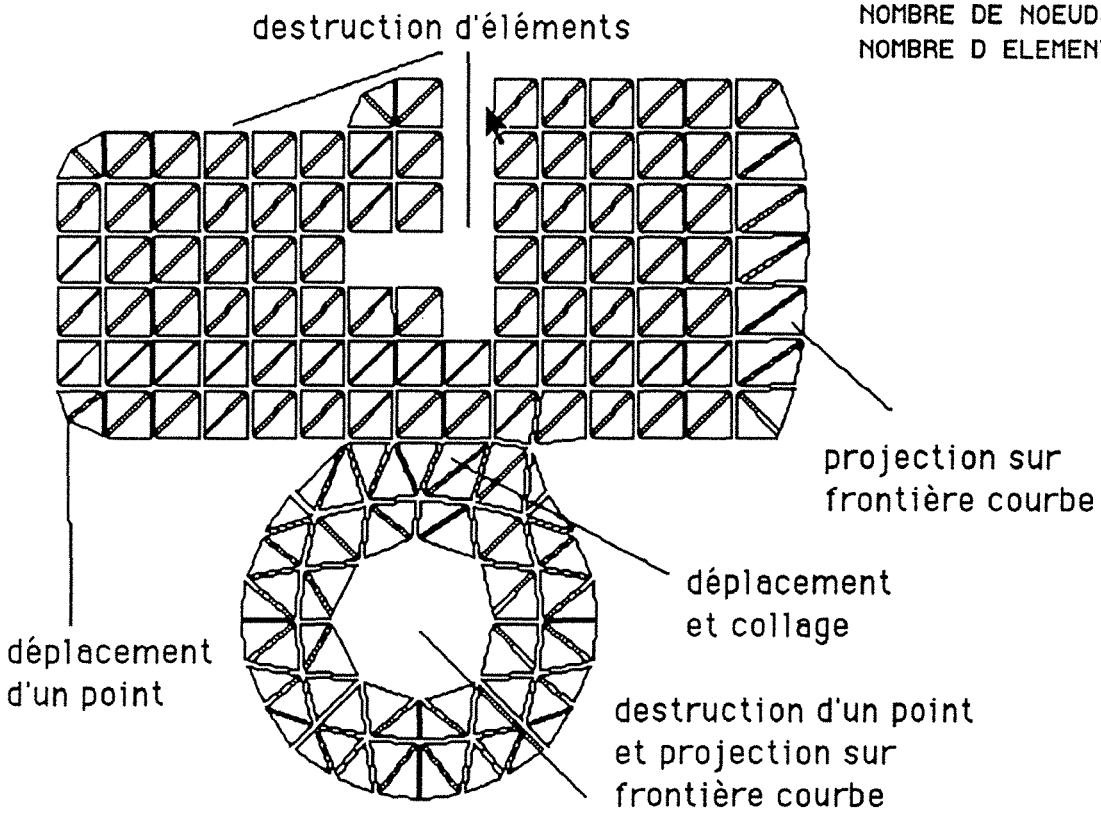


Figure II-26 : opérations effectuées et résultat

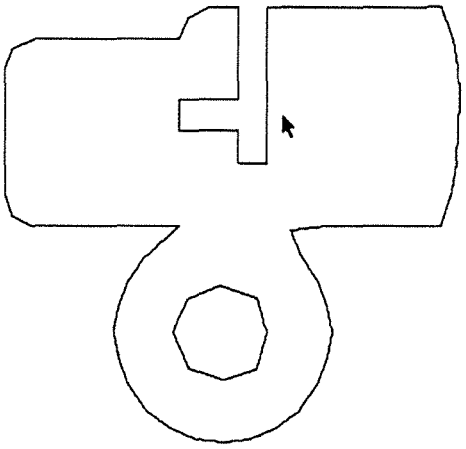


Figure II-27 : contour automatiquement généré

II-7-2 : Maillage d'un rotor de moteur électrique

Cet exemple traite du maillage d'un moteur électrique de géométrie cylindrique. Nous en maillons la représentation plane. Pour des raisons de symétrie, on ne maillera que le 1/6 éme du domaine, mais on verra que l'on peut en déduire l'ensemble, si cela est nécessaire

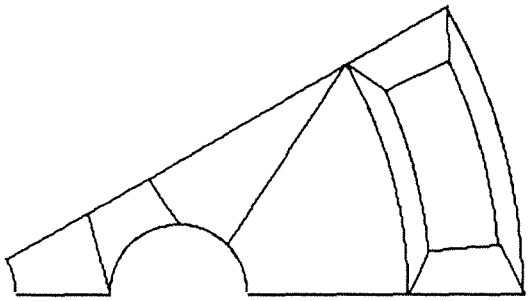


Figure II-28:dessin des blocs
(9 blocs, 40 nœuds)

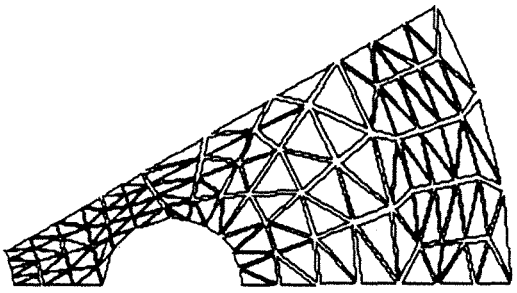


Figure II-29 : dessin du maillage
(138 éléments, 89 nœuds)

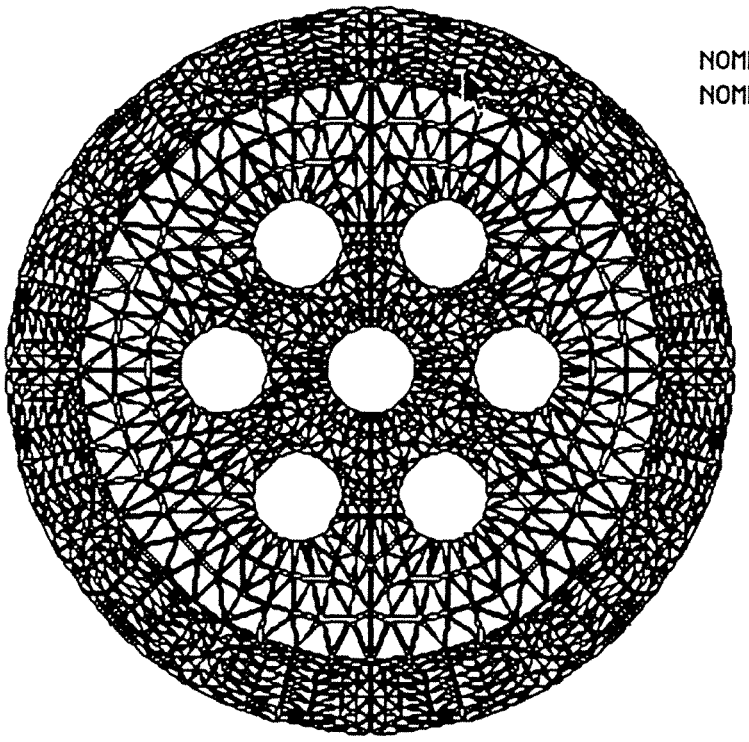


Figure II-30 : maillage complet après une symétrie et 3 rotations
(maillage en triangles 3 nœuds)

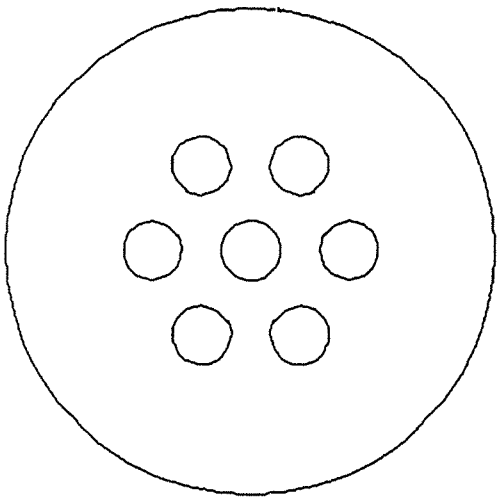


Figure II-31
Contour créé automatiquement

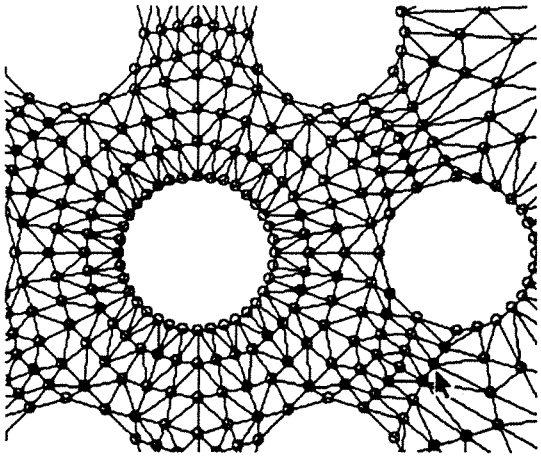


Figure II-32
Zoom du cercle central

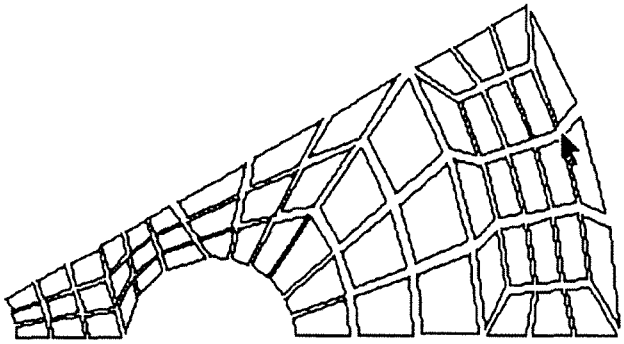


Figure II-33 : maillage d'un secteur
Le même domaine que sur la figure 28, mais maillé en quadrilatères 4-nœuds.

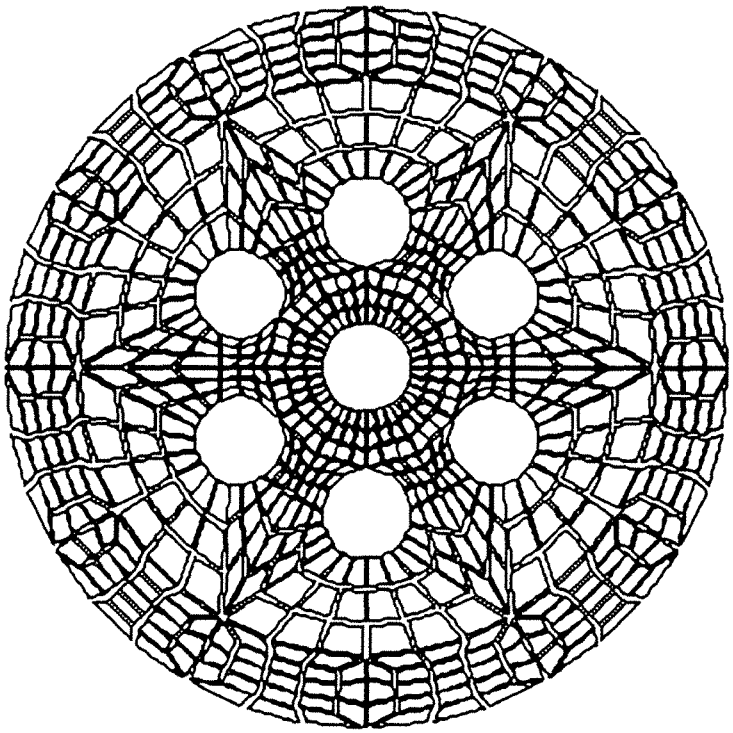


Figure II-34 : maillage total après symétrisation
(684 quadrilatères 4-nœuds et 768 nœuds)

Le temps de calcul de ce problème tout compris (entrée des données suivi du maillage et de la symétrisation) est d'environ 15 minutes sur un Mac II normal.

II-7-3 : Maillages d'implants

On désire mailler un domaine se composant d'un implant circulaire ou d'un implant elliptique, tous deux situés dans un motif rectangulaire. Ce genre de domaine peut se rencontrer dans les problèmes d'homogénéisation. En voici deux exemples :

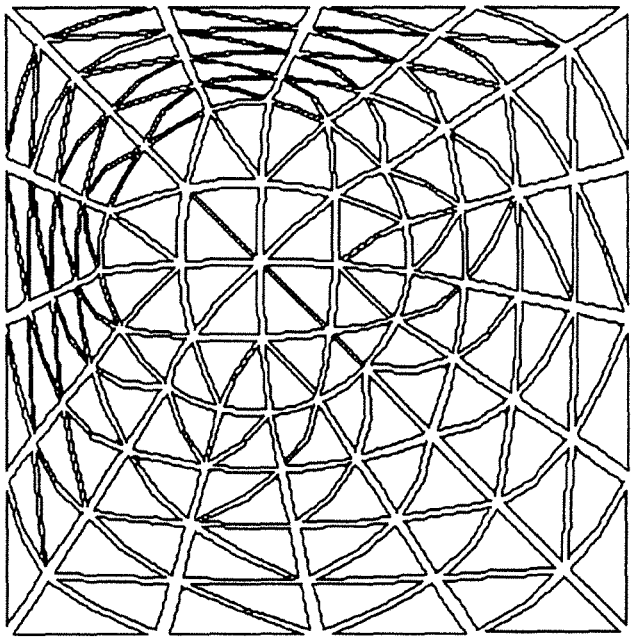


Figure II-35 : Implant circulaire
(160 triangles 6-nœuds et 337 nœuds)

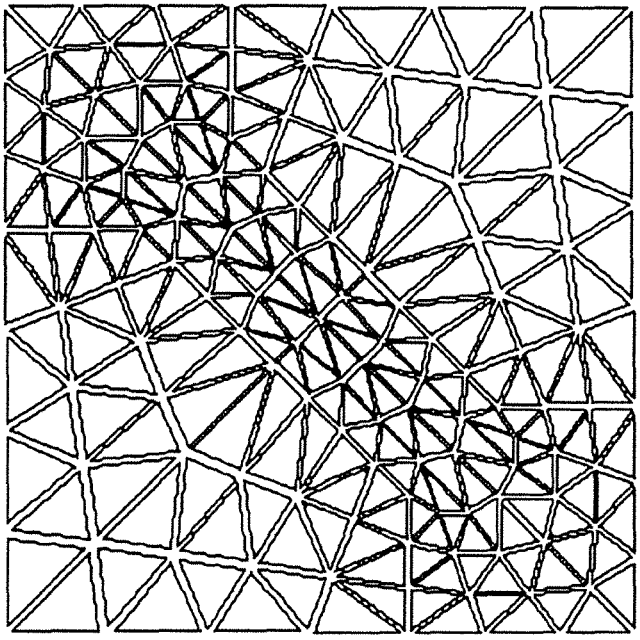


Figure II-36 : Implant elliptique
(236 triangles 3-nœuds et 133 nœuds)

II-7-4 : Maillage d'une clé Anglaise

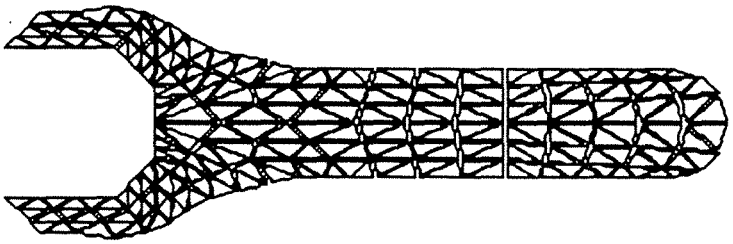


Figure II-37 : maillage d'une clé Anglaise
(240 triangles 3-nœuds et 156 nœuds)

II-8 : Conclusion

Nous avons essayé, dans ce chapitre, de faire une étude exhaustive du mailleur par transformation conforme mis au point, et ce sur plusieurs plans :

-D'abord au niveau de la validité de ce genre de transformation et donc des limites du logiciel.

-Ensuite, au niveau de la structure de données à laquelle le maillage et le contour se réfèrent à tout moment, quelle que soit la modification de l'un ou de l'autre.

-Enfin, au niveau des performances de maillage et des modifications possibles des résultats obtenus.

Ces trois aspects fondamentaux permettent de définir complètement le logiciel, tant au niveau des méthodes que des performances. Le second aspect est certainement le plus important des trois. Il permet au logiciel d'évoluer dans une structure de données homogène et parfaitement compatible avec les programmes de résolution ou les autres programmes de maillage. Ceci permet au logiciel de fournir un résultat utilisable pour la résolution d'autre types de problèmes.

Un des autre intérêt de ce logiciel est sa facilité d'emploi (du moins, nous l'espérons). Ceci est le résultat d'une gestion facilité par l'emploi exclusif de menus déroulant utilisant des saisie graphiques directes à l'écran. A tous niveaux, cette gestion est facilitée par des représentation graphiques (zoom, shrink, ...etc) permanentes, par une série de gardes-fous systématiques limitant au maximum les erreurs possibles, et enfin, par une remise en ordre systématique des données.

L'ensemble du programme compte environ 8000 instructions de calcul. Nous espérons donc avoir obtenu un logiciel efficace et souple d'emploi.

Chapitre III

Maillage automatique par une méthode de Delaunay

III-1 : Introduction

La méthode de Delaunay (comme la méthode de Voronoï, qui est sa duale au sens des graphes, voir Annexe 2), apparait d'emblée parmi les méthodes de maillage les plus prometteuses, et ce à plusieurs titres . D'abord, elle est applicable sans restriction théoriques en dimension n, ce qui permet son utilisation aussi bien en bidimensionnel qu'en tridimensionnel . Elle est relativement aisée à mettre en œuvre et elle n'a pas besoin d'être "adaptée" à chaque problème . Enfin, elle fournit un résultat compatible au sens des éléments finis et de qualité tout à fait acceptable.

C'est donc cette méthode que nous avons choisie d'utiliser pour pouvoir présenter un logiciel de maillage bidimensionnel en triangles ou en quadrilatères totalement automatique . Nous y avons mis en pratique les plus récents développement en matière de maillage afin de pouvoir présenter un produit sûr et efficace.

Comme précédemment, nous nous sommes efforcé d'ajuster le mieux possible les algorithmes entre-eux de façon que le programme utilisé sur micro-ordinateur donne un bon compromis entre la rapidité, l'efficacité et la sûreté . Ce dernier point est de plus renforcé par l'usage constant de menus déroulant rendant le logiciel particulièrement interactif .

III-2 : Description de la méthode

III-2-1 : Notations

En dimension 2, la méthode de Delaunay permet de construire, à partir d'un semis de points donnés, une triangulation complète (Par triangulation, nous entendons ici un maillage en éléments triangulaires 3-nœuds compatible au sens des éléments finis) .

L'un des grand avantages de la méthode réside dans sa totale automaticité . Elle est par ailleurs simple à mettre en œuvre du point de vue théorique quoique son application pratique soit plus délicate . Enfin, comme nous le remarquerons par la suite, cette méthode s'applique tout aussi bien en dimension 3 . [IN 1] [IN 2] [EL 9]

Avant de passer à l'exposé de la méthode présentons quelques notations :

- Nous partons de N points donnés, connus par leurs coordonnées ((x_i,y_i) i=1...N) exprimées dans un repère fixé .

- A partir de ces points, nous allons construire des triangulations en incluant successivement chaque point jusqu'à obtenir une triangulation englobant tous les points (par une méthode que nous allons exposer sans tarder) .Le passage d'une triangulation à une autre se fait par des créations et des destructions de triangles

- Nous noterons T_i le i ^{ème} triangle de la triangulation courante et C_i le cercle circonscrit à T_i .

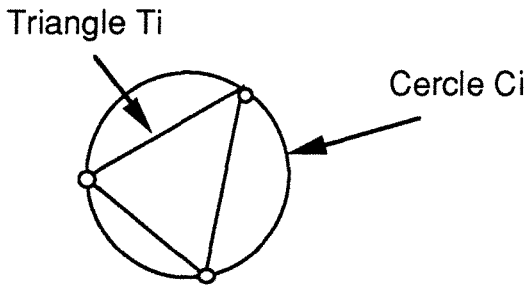


Figure III-1

- Le triangle T_i sera repéré par ses trois sommets décrits dans le sens trigonométrique .

III-2-2 : Méthode classique

Nous disposons de N points fixés au départ, numérotés de 1 à N . Nous construisons la triangulation comme suit:

1 ère étape: On construit le premier triangle avec les trois premier points . Si ces points sont alignés, on permute le troisième point avec le premier point non aligné avec les deux premier .

2 ème étape: On inclut les points restant (de 4 à N) un à un dans la triangulation . La façon dont cette inclusion est faite dépend d'un test .Le résultat de ce test conduit à trois cas possible, incompatibles deux à deux, mais couvrant l'ensemble des cas possibles . Voici ces trois cas :

-a) Le point courant est dans l'un des triangles T_i (et donc a fortiori dans l'un des cercles C_i)

-b) Le point courant n'est dans aucun triangle T_i, mais il se trouve dans un ou plusieurs cercles C_j

-c) Le point courant ne se trouve dans aucun triangle T_i, ni dans aucun cercle C_i

Une fois le point inclus, on passe au point suivant jusqu'à épuisement des points .

La figure III-2 illustre ces trois cas possibles :

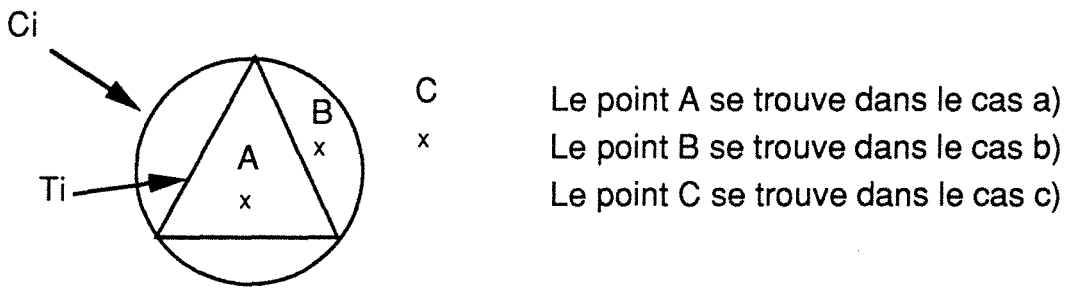


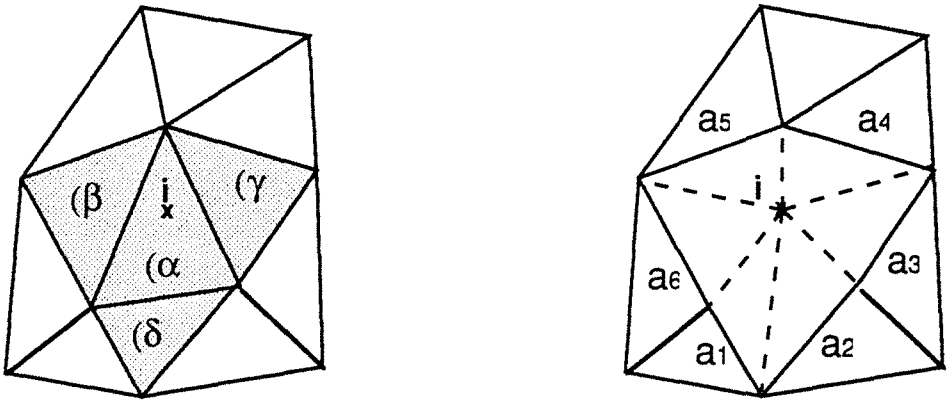
Figure III-2 : les différents cas possibles

Il est à noter que cette méthode nécessite au moins trois points et que l'ensemble des points ne soit pas aligné .

Revenons maintenant en détail sur le mécanisme d'inclusion cas par cas :

Dans le cas a)

Il existe un triangle T_i contenant le points courant . Dès lors, on détermine l'ensemble des éléments dont le cercle C_j contient le point . Ces éléments sont localement détruits, puis de nouveaux éléments sont crées de façon à connecter le point courant au reste du maillage . (exemple figure III-3)



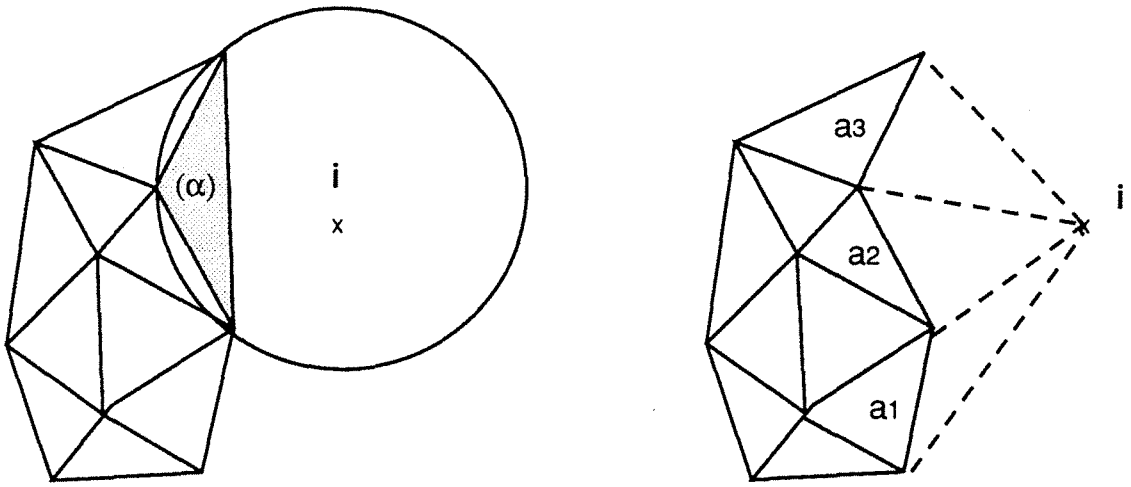
situation avant inclusion
le point i appartient aux cercles circonscrits Cα, ... ,Cγ

situation après inclusion
le point i est relié aux arêtes a1,...,a6 pour créer les nouveaux éléments .

Figure III-3 ; inclusion de points dans le cas a)

Dans le cas b)

Comme précédemment, on détermine l'ensemble des triangles dont le cercle circonscrit C_j contient le point courant, puis on procède comme indiqué sur la figure III-4 :



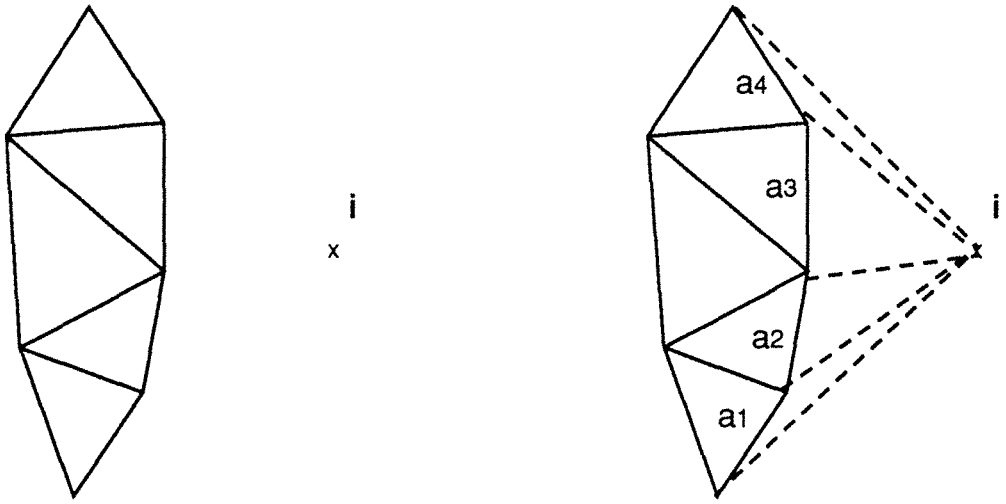
Situation avant inclusion
le point i appartient au cercle Cα

Situation après inclusion
le point i est relié aux arêtes a1, a2,a3

Figure III-4 ; inclusion de points dans le cas b)

Dans le cas c)

On procède comme indiqué sur la figure III-5 :



Situation avant inclusion

Situation après inclusion
liaison du point i avec les arêtes $a_1,\ldots,a_4$

Figure III-5 : inclusion de points dans le cas c)

Notons la grande simplicité des test à effectuer (inclusions dans des triangles ou dans des cercles) . Si l'on compare la complexité de chaque cas, on constate que le cas b) est le plus lourd à traiter, suivi du cas c) puis du cas a) (Ce pour un nombre d'éléments existant n'étant pas trop faible) . Lorsque le nombre d'éléments augmente, cette classification devient encore plus nette ... D'ou l'idée d'un algorithme modifié [IN 1] [IN 2] .

III-2-3 : Méthode modifiée

L'idée fondamentale est basée sur la remarque précédente : on va essayer de manœuvrer de façon à se trouver toujours dans le cas a) . Ceci permet, d'une part, de simplifier la programmation, et , d'autre part, d'améliorer la rapidité de l'algorithme . Enfin, et nous en reparlerons plus loin, ceci augmente la stabilité numérique du processus .

Voyons maintenant le processus en détail

La première étape consiste à construire la boite rectangulaire, de côtés parallèles aux axes, englobant l'enveloppe convexe des N points donnés. Les quatre points sommets du rectangle ainsi obtenu sont légèrement décalés vers l'extérieur de façon à éviter de tomber sur des points existant déjà, ce qui augmente la stabilité de la méthode (figure III-6)

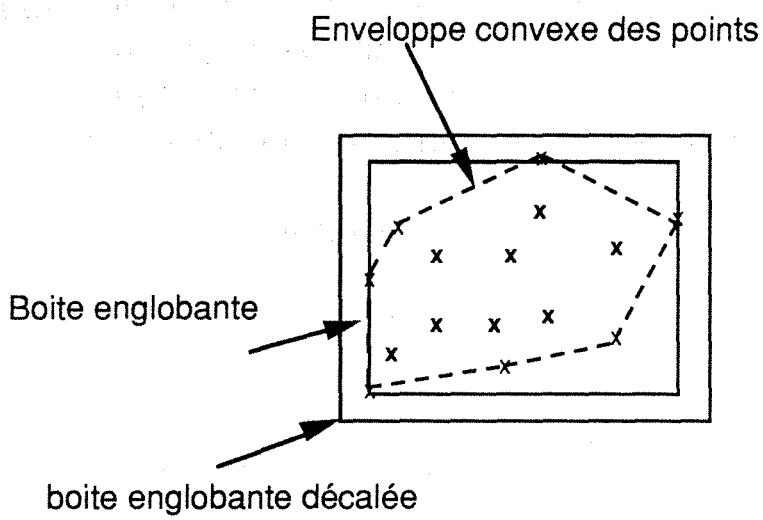


Figure III-6 ; point de départ de la méthode modifiée

Il y a donc création de quatre points supplémentaires: les quatre sommets de la boite. La boite obtenue est divisés en deux triangles suivant la diagonale. On a donc maintenant à inclure les N points de départ en utilisant la méthode classique en partant du maillage de la boite.

Quoi qu'il advienne par la suite, on voit de façon évidente que tout point à inclure sera forcément dans le cas a) . Une fois le maillage effectué en totalité, on n'aura plus qu'a éliminer les éléments dont l'un des sommets fait partie de l'un des quatre sommets de la boite englobante de départ.

D'un point de vue pratique, cet algorithme conduit à des résultats quasi identiques à ceux obtenus par la méthode classique .

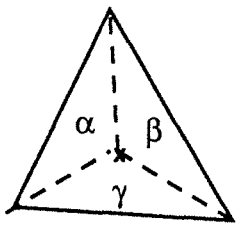
III-2-4 : Remarques générales (quoique indispensables !)

La méthode de Delaunay converge numériquement pour donner un maillage compatible au sens des éléments finis pour peu que l'on fasse attention à plusieurs aspects; sans quoi l'ordinateur se fera un plaisir de donner absolument n'importe quoi ! .

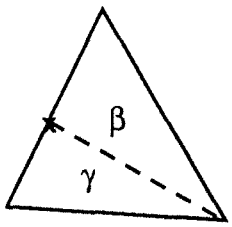
1- Il est absolument indispensable que la méthode soit utilisée de manière indivisible et non combinée avec d'autres méthodes . Relier plusieurs blocs de maillage entre-eux par la méthode est certes faisable, mais extrêmement périlleux .De façon plus générales, si entre deux inclusions de points, on modifie le maillage, il n'est plus assuré que ce dernier reste cohérent (problèmes de chevauchements d'éléments entre-autres) . L'ensemble de la méthode de Delaunay forme donc un tout que l'on ne peut dissocier sans précautions .

2-Une autre remarque liée à la précédente.. La méthode génère automatiquement un maillage formant un polygone convexe .Si un maillage obtenu par une méthode donnée ne forme pas un domaine convexe, il ne peut être issu de la méthode de Delaunay, ni être utilisé sans précaution avec cette dernière . Dans le cas contraire, la frontière polygonale du maillage obtenu coincide avec l'enveloppe convexe de l'ensemble des points . Sans anticiper sur la suite, il est à prévoir que si un domaine à mailler n'est pas convexe, des corrections seront à effectuer pour répondre aux exigences du problème posé .

3- Après chaque inclusion de points, il est nécessaire de tester la présence éventuelle d' "éléments d'aire nulle" . de tels éléments peuvent apparaître dans le cas de points alignés, comme le montre la figure III-7 .



a) cas normal
création de trois triangles α, β , γ



b) cas dégénéré
le triangle α ayant une aire nulle doit être supprimé

Figure III-7 ; cas dégénérés

Même s'ils sont théoriquement bien alignés, numériquement, les points forment un triangle dont l'aire n'est pas identiquement nulle, mais plutôt de très faible valeur . Il s'en suit que le rayon du cercle circonscrit correspondant est énorme (théoriquement, il est infini) .Si un tel élément n'est pas éliminé tout de suite, il fera diverger l'algorithme de construction du maillage .

4-Pour finir; une remarque d'ordre général: la méthode de Delaunay se généralise à d'autres dimension, notamment en dimension 3 . Il suffit dans l'exposé de remplacer les triangles par des tétraèdres et les cercles circonscrites par des sphères circonscrites à chaque tétraèdre . L'inclusion de chaque point se fait par liaison avec des faces triangulaires et non plus avec des arêtes . Sinon le déroulement de l'algorithme reste le même .

III-2-5 : Problèmes des stabilité . Convergence de la méthode

La méthode de Delaunay peut présenter des problèmes de stabilités liés essentiellement à des incertitudes numériques (dues aux positionnement des points) lors de l'évaluation de certains "cas tangents" . L'exemple des triangles à "aire nulle" est une illustration de ce genre de problème ; Nous avons vu comment on pouvait pallier à ce genre d'inconvénients .

Toujours dans le même ordre d'idée, de graves problèmes peuvent apparaître lors de l'inclusion de points co-circulaires .

Prenons l'exemple de la figure III-8 :

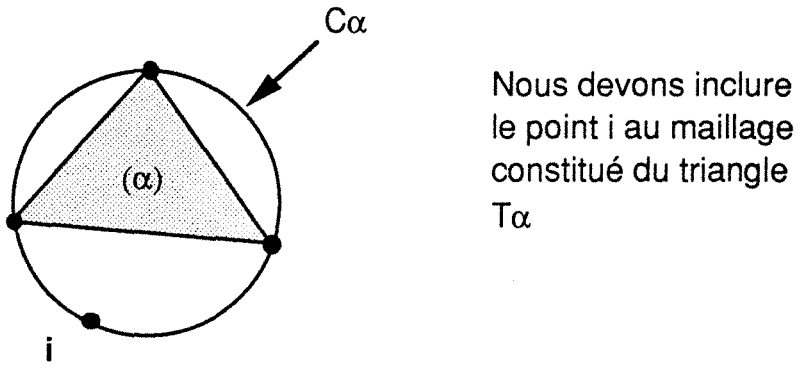
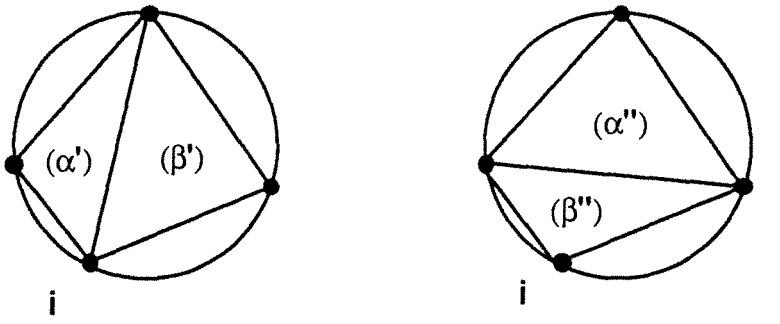


Figure III-8

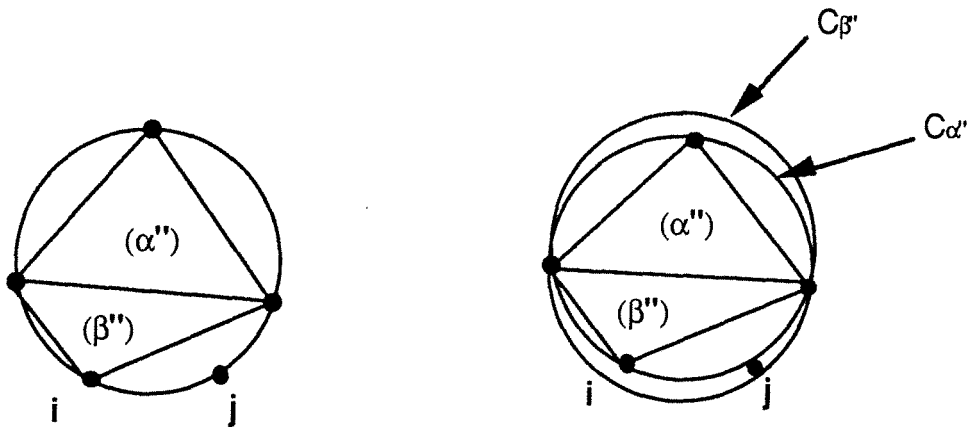
D'un point de vue théorique, le point **i** est situé juste sur le cercle C_α . D'un point de vue pratique, selon la précision numérique des test ainsi que des coordonnées, le point **i** pourra se trouver à l'intérieur de C_α (cas b)) ou à l'extérieur (cas c)) . Cependant, dans les deux cas, le maillage reste correct (figure III-9) .



Maillage dans le cas b) Maillage dans le cas c)

Figure III-9

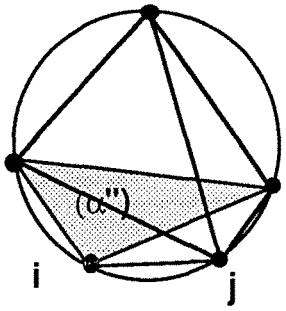
Maintenant, à partir du cas c), incluons un nouveau point j co-circulaire avec C_α Pour des raisons d'arrondis numériques, lors de l'évaluation des tests, on trouve (à ε près) que le point j appartient à $C_{\beta"}$ mais pas à $C_{\alpha"}$ (figure III-10) .



Situation théorique Situation réelle

Figure III-10

Lors de l'inclusion du point j, on tombe sur le résultat suivant:



Résultat du maillage

Le triangle β " a disparu. Plusieurs des triangles nouvellement crées coupent le triangle α " : situation incorrecte .

Figure III-11 ; exemple de maillage incorrect

Il est cependant à noter que si le résultat du test auvait été légèrement différent, cela aurait pu conduire à des résultats corrects .

Nous constatons donc que des difficultés peuvent apparaître lord de l'évaluation de points "presque co-circulaires" . Par point "presque co-circulaire", nous entendons un point se trouvant légèrement (en valeur relative) à l'intérieur ou à l'extérieur du cercle correspondant . Pour lever l'indétermination due au calcul, plusieurs solutions sont possibles ([IN 1] [IN 2]) :

- augmenter la précision des calculs
- rejeter à la fin l'inclusion des points critiques en espérant que l'ambiguïté aura été levé par l'inclusion des autre points .
- légèrement déplacer les points critique de façon à se retrouver dans un cas normal, puis replacer les points après maillage dans leur position initiale
- évaluer les tests à partir de coordonnées entières [IN 2]

Chacune de ces méthode a ses limites .Ainsi, la première ne fait que repousser plus loin l'échéance de la décision . La seconde est plus sûre, mais la convergence ne peut être systématiquement assurée . La troisième est limitée par le fait que certains points sont imposés (points du contour) et ne peuvent être déplacés. La quatriëme semble la méthode la plus sûre, pour peu que l'on puisse s'y ramener.

Nous avons donc choisi d'appliquer la troisième méthode pour son bon rapport efficacité/complexité . En ajoutant à cela une étude préalable du diamètre de l'enveloppe convexe des points permettant un ajustement des seuils critiques, on obtient une méthode fiable (mais pas à 100% cependant).

En dimension 3, toutes les méthodes et remarques vues précédemment restent valables, mais les inconvénients signalés apparaissent plus nettement . Ces questions sont encore loin d'être résolues puisque de nombreuses études sont encore en cours pour générer des algorithmes permettant de lever les indéterminations dans les cas critiques, notamment [IN 1] et [IN 2] .

III-3 : Principe de maillage automatique

III-3-1 :Introduction

Il est naturel d'associer la méthode de Delaunay à un processus de maillage automatique en utilisant les points du contour, pour générer une triangulation correcte . Cependant, il est possible qu'une telle triangulation soit de mauvaise qualité (éléments très aplatis, éléments de tailles très différentes) . Une seule "passe" de maillage n'est donc pas suffisante . On doit effectuer d'autres "passes" destinées à ajouter de nouveaux points au barycentre d'éléments jugés trop gros. On itére ainsi jusqu'à obtenir un maillage "homogène" .

Examinons maintenant l'ensemble du mécanisme de maillage automatique, tant au niveau de la structure de données qu'au niveau purement arithmétique des décisions .

III-3-2 : Structure de données

La première chose à définir est le contour externe du domaine . Ce contour doit être fermé et sans points doubles . Il se compose d'un ensemble d'arêtes droites ou circulaires, elles-mêmes se composant chacune de deux points (plus quelques informations nécessaires aux arêtes courbes: rayon, centre etc ...) . Ces arêtes composant le contour sont appelées "arêtes externes" . Elles permettent, entre autre, de définir l'intérieur de l'extérieur du domaine . Leur structure est celle décrite dans le chapitre I .

Ces arêtes devant être classées bout à bout dans le sens du parcours du contour (sens trigonométrique ou non, peu importe) . Ce classement est effectué automatiquement par le logiciel qui teste au passage la validité du domaine rentré .

Ce type de structure permet de traiter tous les types de non-convexité, sauf celles occasionnées par des trous . Nous avons donc décidé d'adjoindre aux arêtes extérieures d'autre arêtes appelées "intérieures" par opposition aux précédentes . Ces arêtes sont, géométriquement parlant, du même type que les précédentes et servent à forcer le maillage à s'appuyer sur certaines lignes données (ce qui permet par exemple de simuler un changement de milieu) . Le problème des trous est plus ardu à traiter . Il est possible d'incorporer tout trou au contour externe par l'intermédiaire d'une coupure décrite dans les deux sens (Figure III-12) .

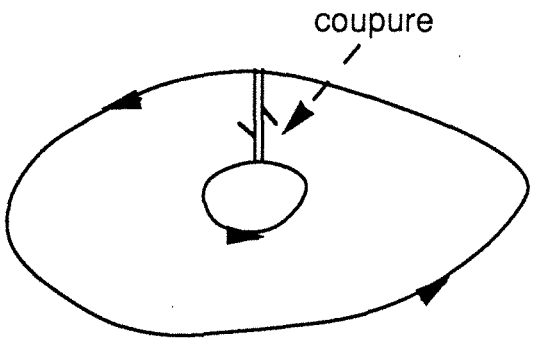
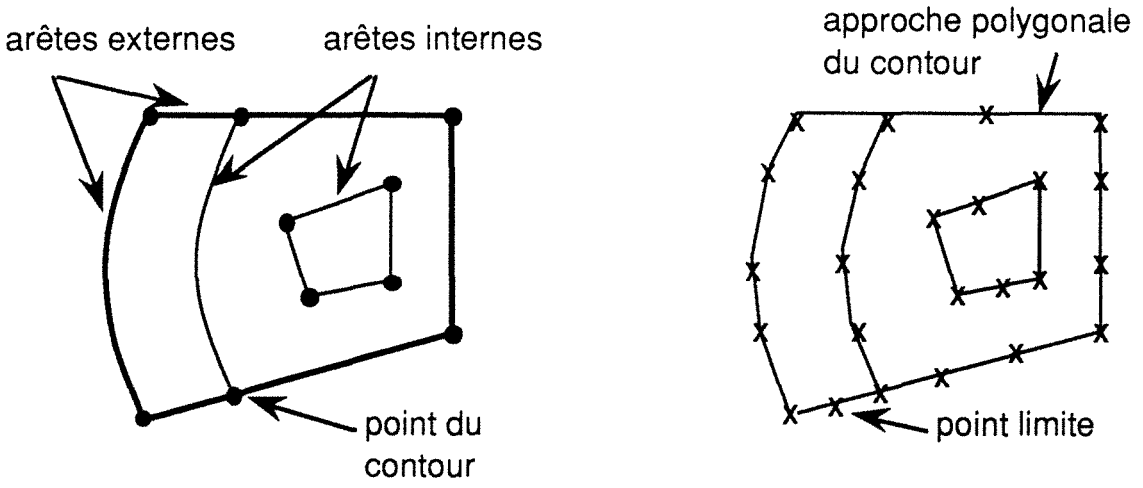


Figure III-12

Cependant, la méthode de Delaunay maillera le trou car elle conduit à un maillage convexe quoi qu'il advienne . Nous avons donc décidé de traiter les trous du domaine comme des lignes internes . Une fois le maillage effectué, il sera possible d'éliminer les éléments à l'intérieur des trous .

La description de toutes les arêtes nécessite la connaissance d'un certain nombre de points que nous appelerons "points du contour"

L'ensemble (points du contour-arêtes) permet de définir complètement le problème . Cet ensemble est stocké en fichier externe pour permettre une utilisation ultérieure du domaine . A ces points s'ajoutent de points dits "points limites", issus de la division de chaque arête en un certain nombre de segments de droite . Ces points correspondent au densités d'éléments que l'on veut avoir sur les bords . Comme pour le mailleur par blocs, il est possible d'avoir des répartitions linéaires ou semi-logarithmiques centrées sur un point . Ces densité sont réglables et modifiables sur chaque arête individuellement, à la demande de l'utilisateur.



contour interne et externe

division des arêtes et position de départ du mailleur

Figure III-3 ; un exemple de données

On notera que cette structure de données est plus souple et plus facile d'emploi que celle nécessaire au mailleur par bloc . Par ailleurs, les divisions d'arête peuvent être prises comme il semble bon, il y a donc un progrès vis à vis de la facilité d'emploi . Une facilité qui se trouve accrue par la rentrée automatique du contour par saisie directe à l'écran . Cette rentrée se décompose en deux étapes

- entrée des "points du contour" . Cette étape est facilitée par de nombreux utilitaires (symétrie, translation etc ...) .

- entrée des arêtes ; Là encore, une série d'utilitaires utilisant des fonctions de CAO facilitent le travail (calcul automatique de centres, raccord automatique par tangence...).

La gestion du contour étant en grande partie automatique, nous avons mis en place une série de contrôles évitant au mieux les aberrations, dans l'intérêt de l'utilisateur .

III-3-3 : Ajout de points et subdivision de maillage

A partir des informations précédentes, nous devons mailler un domaine à partir d'un contour polygonal approchant son contour . Une fois le maillage effectué uniquement à partir des points situés sur la peau, nous allons adjoindre un point au barycentre des éléments "jugés trop gros" . Ces points seront inclus au maillage par un processus de Delaunay .

Mais sur quel critère se baser ? . L'idée la plus naturelle serait de comparer l'aire de chaque éléments à une aire moyenne issue de la donnée des divisions sur le bord du domaine . Mais un tel procédé peut générer des ajouts de points intempestifs menant à une mauvaise convergence du processus et à la création d'éléments de plus en plus mauvais . C'est particulièrement le cas pour les éléments ayant un côté sur le contour .

Figure III-14 ; exemple de mauvaise convergence

Dans l'exemple ci-dessus, l'élément (α) ayant une arête frontière (imposée par l'utilisateur) de longueur très au dessus de la moyenne, est donc divisé . Ceci crée donc trois éléments de mauvaise qualité . En particulier, l'élément (β) a sa concavité tournée vers l'extérieur, il a donc toutes les chances de demeurer ainsi au cours du processus de maillage, n'interférant avec aucun points .

Pour pallier ces inconvénients, nous proposons un test différent répondant mieux aux densités imposées sur les bords . Il s'effectue comme suit:

- soit l'élément courant a au moins une arête sur le bord . Dans ce cas, la taille de l'élément est comparée à la taille d'un triangle fictif équilatéral construit à partir de cette arête . S'il y a plus d'une arête frontière, on ne prend comme référence que la plus grande .

- soit l'élément courant est interne . Dans ce cas, on compare l'aire de l'élément à l'aire moyenne des élément du maillage .

Ce procédé a l'avantage de fournir un maillage sans disparités et s'adaptant bien à des densités très différentes (particulièrement pour les densités logarithmiques) . Une seule précaution à prendre: pour assurer la qualité du maillage dans le cas de domaines non convexes, nous décidons de ne pas diviser les éléments extérieurs au contour apparus lors du processus de Delaunay .

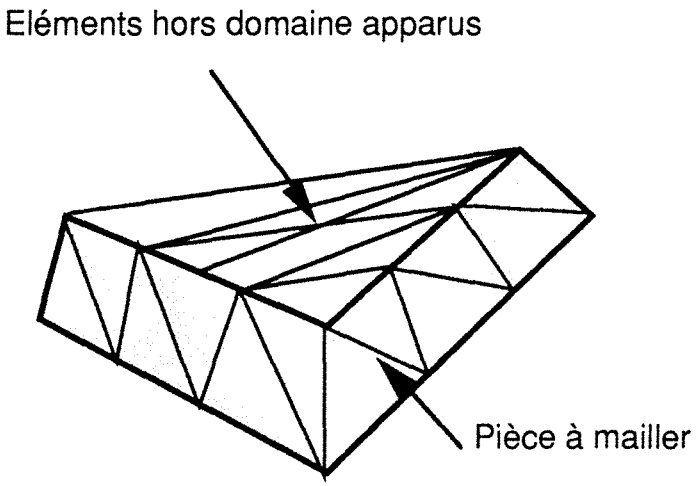


Figure III-15 ; maillage d'une pièce non convexe

Ces éléments sont généralement de mauvaise qualité, comme on peut le remarquer sur la figure III-15 . Pour des raisons de simplicité de la méthode et surtout de qualité de maillage, ces éléments ne sont pas divisés.

III-3-4 : Test de convexité

D'après ce que nous venons de dire, il est très rentable de déterminer si le domaine est convexe ou non (nous rappelons que dans une première étape, nous ignorons les trous du domaine) . Nous travaillons sur le contour polygonal approchant notre domaine . Si le domaine est convexe, son contour polygonal l'est aussi . Le contour polygonal correspond à l'enveloppe convexe de ses points . Donc tout maillage par Delaunay ne crée aucun éléments extérieur au contour, il n'est alors pas nécessaire de tester l'appartenance des éléments au contour, d'ou un gain de temps évident .

Il est facile de déterminer la la convexité d'un contour polygonal par le suivi segment par segment du contour . Ce suivi doit être fait bout à bout . On effectue le produit scalaire entre deux segments consécutifs sur tout le contour . Si le produit vectoriel change de signe au moins une fois, le contour est concave, sinon il est convexe .

Le suivi du contour arête par arête permet aussi de déterminer l'intérieur du domaine de l'extérieur ; Pour cela, nous utilisons la méthode de la section par une demi-droite . Pour savoir si comment un point par rapport au domaine, nous construisons une demi-droite quelconque issue de ce point et partant à l'infini, puis nous comptons le nombre d'intersections entre le contour et la demi-droite . Si ce nombre est pair, il est à l'extérieur, s'il est impair, il est à l'intérieur (figure III-16) .

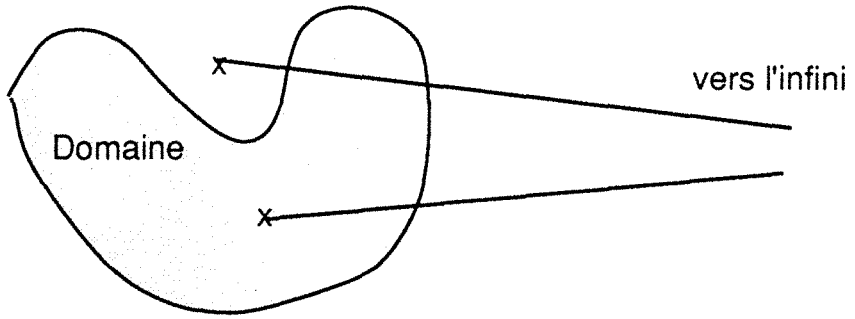


Figure III-16 ; méthode de la demi-droite

C'est là un test fiable et convenant à tous les types de contour . Il convient cependant d'éviter de se trouver dans un cas critique (figure III-17)

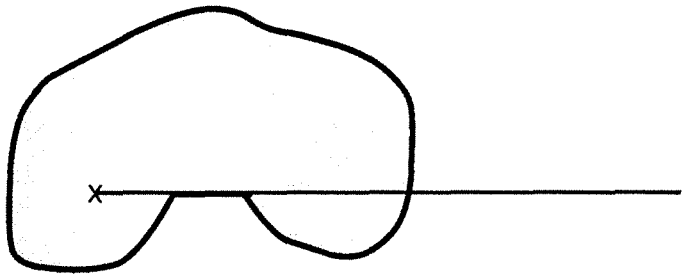


Figure III-17 ; exemple de situation critique

A ces cas critique s'ajoutent des "cas douteux" dus aux incertitudes numériques sur les coordonnées . Pour éviter cet écueil, nous procédons comme suit:

> Parti d'un point à l'infini fixé et du point à tester, nous vérifions qu'aucun point du contour (polygonal) n'appartient à la demi-droite. Si ce n'est pas le cas, nous faisons pivoter la demi-droite de quelque degrés jusqu'à ce soit le cas. Nous sommes sûrs dès lors de ne plus nous trouver dans un cas critique .

III-4 : Problème de non convexité et forçage d'arêtes

III-4-1 : Introduction

Comme nous l'avons vu précédemment, la méthode de Delaunay ne maille que l'enveloppe convexe de l'approximation polygonale d'un domaine . Il y peut donc y avoir des éléments extérieurs à éliminer . Dans le cas où ceux-ci sont complètement extérieurs, il est facile de les détecter (par une méthode vue au paragraphe précédent) et de les détruire, comme le montre la figure III-18 :

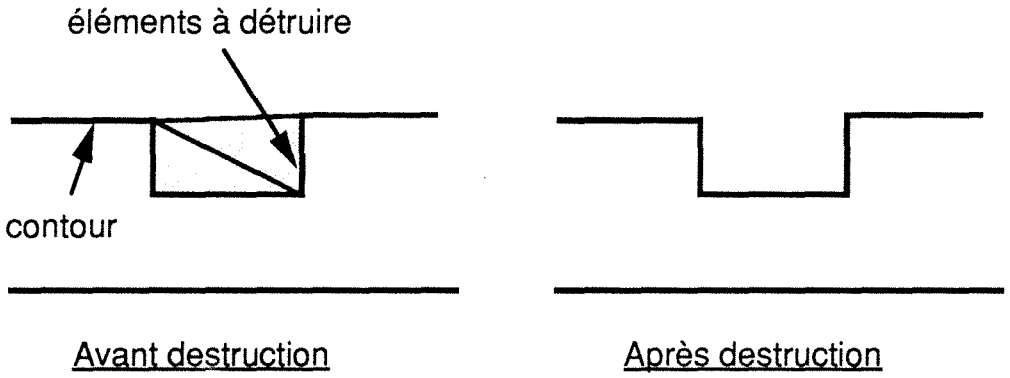
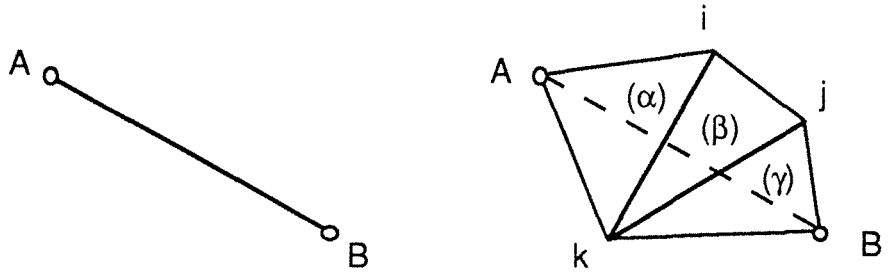


Figure III-18

Il existe cependant des cas où les éléments ne sont ni totalement à l'intérieur ni totalement à l'extérieur du domaine et ne peuvent donc pas être éliminés sans précautions .

Analytiquement, cela se traduit par le fait qu'au moins une des arêtes du contour polygonal ne se retrouve plus dans le maillage existant, situation schématisée par la figure III-19 :



Situation de départ
l'arête AB est imposée

Après maillage
l'arête AB ne se retrouve plus

Figure III-19 : disparition d'arêtes lors du maillage

Pour pallier ce genre d'inconvénients, il a été développé une technique [IN 2] permettant de substituer localement au maillage ne défaut un maillage équivalent mais contenant l'arête incriminée : le Forçage d'arêtes . En voici l'exposé .

III-4-2 : Principes de base . Notions de maillages équivalents

Nous nous proposons de mailler une approximation polygonale d'un domaine défini par ses arêtes (internes et externes) .

Définition: respect du contour

Nous dirons qu'un maillage respecte exactement son contour polygonal si il contient de manière exacte parmi ses arêtes tous les segments de ce contour .

De façon générale, une triangulation de Delaunay ne respecte pas le contour systématiquement . C'est particulièrement le cas lorsque la domaine n'est pas convexe (cas de fissures par exemple) . Dans ce cas les points voisins interfèrent entre-eux causant un non-respect du contour .

Définition: maillages équivalents

Soient deux triangulations T_1 et T_2, compatibles au sens des éléments finis . Nous noterons $D_1 = \boldsymbol{U}\, t_i$ et $D_1 = \boldsymbol{U}\, t'_j$, t_i et t'_j étant les éléments de T_1 et T_2 respectivement

Nous dirons que les deux triangulations sont égales si :

- *D1=D2*
- *Les contours de D1 et D2 sont discrétisés de la même façon*

Cette notion d'équivalence est purement géométrique (même contour, même domaine) et ne préjuge pas des qualités respectives de chaque maillage . Si la seconde propriété n'est pas vérifiée, on parle de maillage sous-équivalents .

L'illustration la plus simple de cette définition est celle de deux triangles adjacents donnée figure III-20 (inversion de diagonale) . On notera que suivant la position des points A,B,C et D , l'équivalence n'est pas toujours possible .

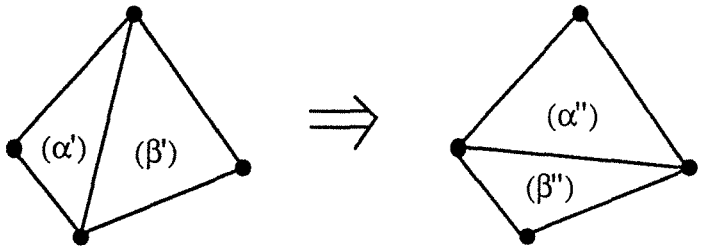


Figure III-20 ; maillages équivalents

A partir de ces définitions, nous allons essayer de construire un maillage équivalent à celui fournit par la méthode de Delaunay, mais respectant exactement le contour .

III-4-3 : Processus détaillé

Considérons une arête du contour (A,B) , arête que l'on ne retrouve plus dans le maillage (figure III-21) . Voici maintenant les étapes de la transformation:

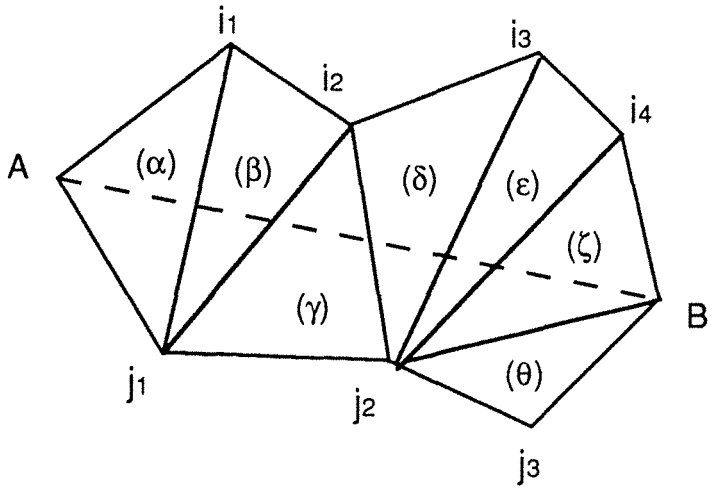


Figure III-21 ; situation avant forçage

1°) : détermination de tous les triangles coupant l'arête (A,B)

Il s'agit ici des triangles (α) ...(ζ) . A partir de là, on déduit l'ensemble des points liés à ces triangles . Ces points se divisent en deux catégories : les points "en dessus" de l'arête (ici $i_1,\ldots,i_4$) et ceux "en dessous" (ici $j_1,\ldots,j_3$) . A noter que les points A et B sont considérés comme étant en même temps en dessus et en dessous .

2°) : suppression de tous les triangles (α) ...(ζ)

Puis nous allons maintenant traiter successivement les points "en dessus" puis "en dessous" de l'arête . Le traitement étant symétrique, nous ne considérerons sur notre figure que ceux "en dessus" .

3°) : traitement des points du dessus

a) détermination du point i le plus proche de l'arête (A,B), point que nous noterons i_0

b) création de l'élément $[A,i_0,B]$ (figure III-22)

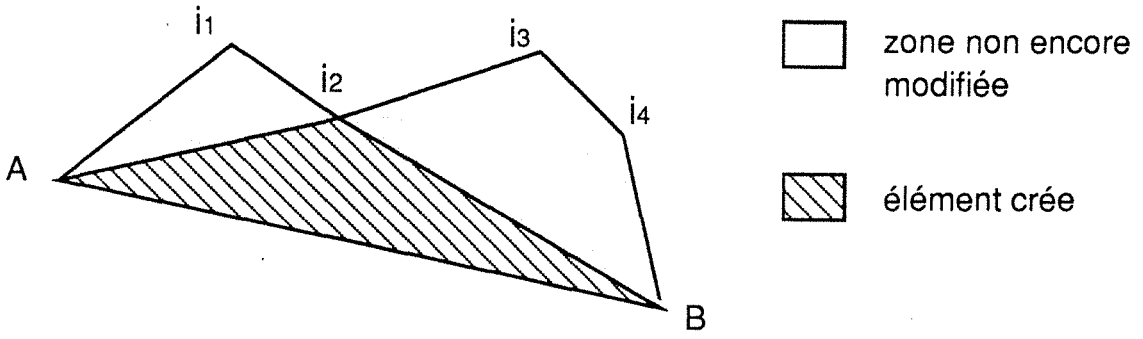


Figure III-22 ; situation en cours de forçage

On peut constater la réapparition de l'arête (A,B) . Nous allons maintenant travailler successivement sur les arêtes (A,i_0) et (i_0,B) . Ces deux problèmes sont de complexité strictement inférieure au problème de départ, ce qui prouve la convergence de l'algorithme .

c) traitements récursifs des arêtes (A,i_0) et (i_0,B) .

Ce travail est effectué jusqu'à épuisement des points i

4°) : traitement des points du dessous

Le traitement est identique à celui des points du dessus .

Après forçage de l'arête, nous obtenons finalement:

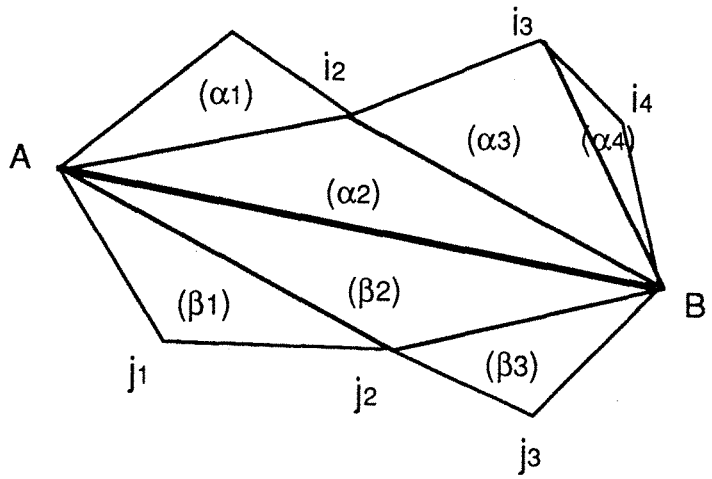


Figure III-23 ; résultat final du forçage

Plusieurs remarques s'imposent à propos du résultat obtenu:

- L'arête (A,B) a totalement réapparu (c'était le but du jeu)

- Le maillage est correct au sens des éléments finis

- Le maillage résultat est parfaitement équivalent au maillage précédent (même domaine, même contour), il n'y a donc aucune adaptation à faire .

- Les nouveaux éléments sont, soit totalement au dessus, soit totalement au dessous de l'arête . L'intérêt de ceci est évident si l'arête (A,B) est une arête externe : on pourra alors éliminer les éléments inutiles sans problèmes .

- On remarquera la moins bonne qualité des éléments après transformation . Ceci sera corrigé par un lissage dont nous reparlerons plus loin

III-4-4 : Utilisation du processus de forçage

Le forçage de toute les arêtes du contour polygonal a deux buts:

1) fournir un maillage respectant exactement le contour
2) permettre d'éliminer sans problèmes les éléments extérieurs au contour .

Nous allons itérer sur chaque segment du contour polygonal:

- Si le segment apparait dans le maillage, on ne fait rien
- Sinon , on force le segment pour le faire réapparaître

III-4-5 : Généralisation et conclusion

La méthode de forçage présente de multiples variantes [IN 2] . En particulier, notons un prolongement intéressant en dimension 3 . Le problème est alors de forcer une face dans un maillage en tétraèdres (par exemple issu d'un mailleur de Delaunay en dimension 3) .La formulation pratique est nettement plus complexe, mais l'esprit de la méthode reste le même . Cette méthode a été testée avec succès sur de gros maillages tridimensionnels fort complexes par les équipes de l'INRIA [IN 2] .

III-5 : Utilitaires divers

Comme pour le mailleur par bloc, il nous est apparu indispensable d'adjoindre des utilitaires de post-traitement permettant d'augmenter les performances du mailleur . Comme d'habitude, nous leur avons imposé d'être les plus conviviaux et les plus sûrs possible . Nous en présentons quatre .

III-5-1 :Lissage de maillage

Il ne s'agit pas à proprement parler d'un utilitaire, mais plutôt d'un contrôle automatique . Après toute modification perturbant le maillage, cet utilitaire est appelé pour redonner au maillage la qualité et le bon équilibre initial .

Le principe en et simple : essayer de trouver une position optimale des points de façon que les éléments soient le plus proche possible du triangle équilatéral, tous les points sur le contour restant fixes bien sûr .

Pour cela, nous utilisons une technique de recentrage de chaque point au barycentre de l'ensemble de ses proches voisins (figure III-24) .

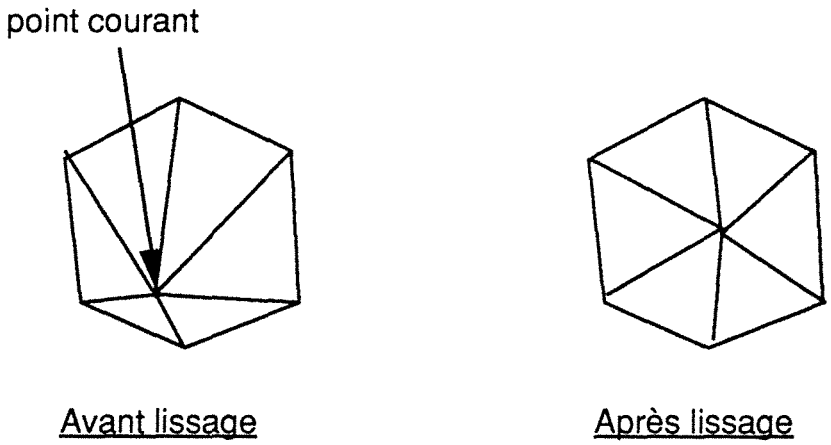


Figure III-24 ; lissage par recentrage

Cela conduit à l'obtention du système NxN (où N est le nombre de points du maillage) à résoudre . Dans la pratique, ce système n'est jamais explicité. On procède par approximations successives en prenant tous les points dans l'ordre et en les déplaçant au barycentre de leurs voisins actuels. Quelques itérations de cette méthode facile à mettre en œuvre et fiable suffisent à obtenir un bon résultat .

Cet utilitaire apporte de façon visible un plus grand équilibre dans le maillage . Cela conduit à de meilleurs résultats lors de la résolution .

III-5-2 :Raffinements locaux de maillage

Il peut être parfois nécessaire de raffiner à la demande une partie quelconque du maillage sans que le reste ne soit modifié . Ceci est particulièrement intéressant lorsque l'on sait que les valeurs étudiés varient fortement dans une zone donnée (zone de concentration de contraintes par exemple) alors que dans le reste du domaine, ces mêmes valeurs varient peu.

Nous avons développé un utilitaire souple d'emploi et interactif qui raffine automatiquement une zone saisie à l'écran . Le processus se divise en deux étapes :

- 1°/ Division en 4 de tous les éléments de la zone .
- 2°/ Division des éléments jouxtant les éléments divisés de façon à assurer la continuité du maillage au sens des éléments finis .

Pratiquement, lors de la première phase les éléments sont divisés conformément à la figure III-25 :

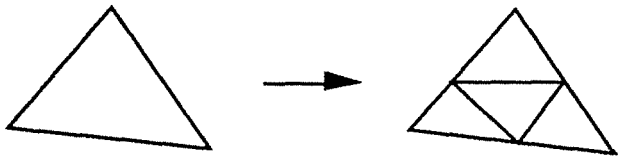


Figure III-25 : principe de division

Lors de la seconde phase, la division est effectuée selon le nombre de voisins divisés au cours de la phase précédente (figure III-26)

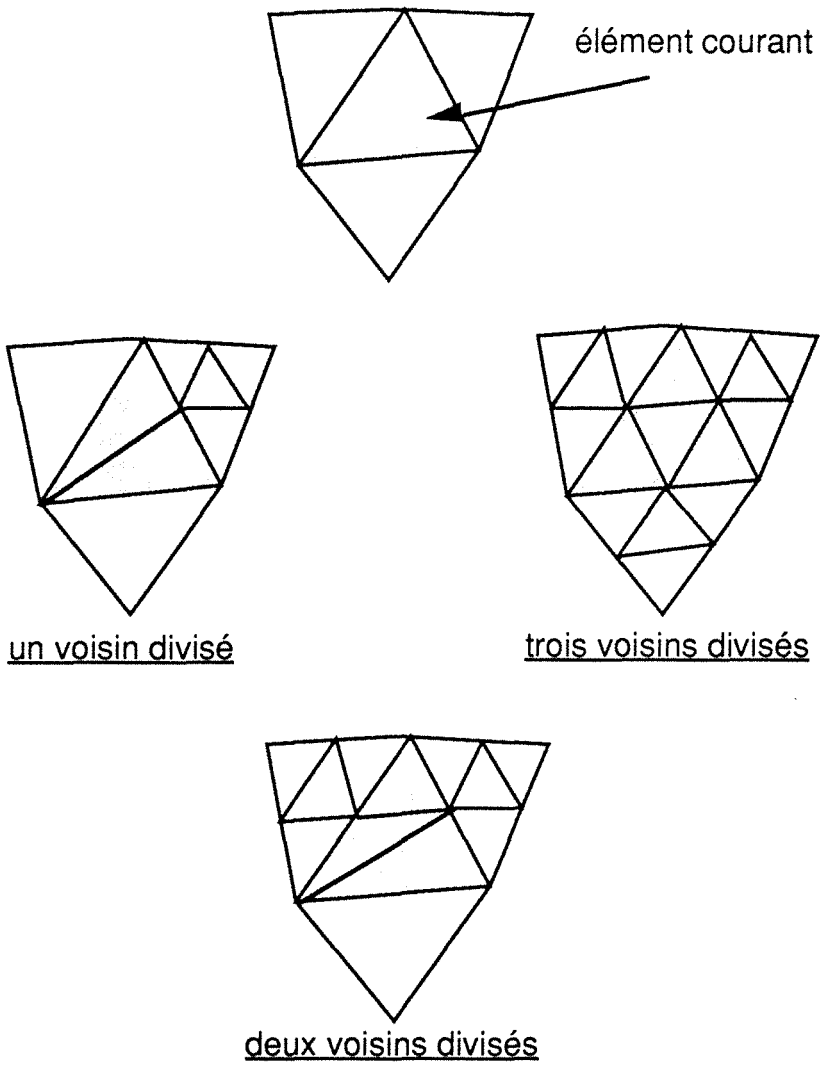


Figure III-26 ; types de raccordement de maillage

Lors du processus, en fait, les deux étapes sont réalisées en même temps, ce qui produit un gain de temps notable . Il est à noter que de nombreuses modifications de maillages interviennent durant le processus . Le programme se charge automatiquement de tout remettre en œuvre pour avoir une structure de données conforme (éléments, côtés frontières etc ...) .

Lors de la division d'un élément ayant un côté frontière situé sur une arête courbe, il y a projection automatique sur cette arête courbe (figure III-27) .

projection sur frontière courbe

Figure III-27

Comme on aura pu le noter dans l'exemple, l'utilitaire peut générer des éléments de qualité médiocre . Cette situation est corrigée par lissage, ce qui redonne un aspect régulier au maillage .

L'ensemble de ces fonctions permettent une ré-utilisation quasi-infinie de cet utilitaire .

III-5-3 :Transformation en éléments quadratiques

Il peut être intéressant de pouvoir augmenter le degrés de l'interpolation liée à chaque élément, par exemple pour avoir plus d'informations à nombre d'éléments constant, ou bien parce que certains problèmes nécessitent des interpolations de degrés plus élevé .

Nous rappelons que le logiciel maille en triangles 3-nœuds, qu'il peut aussi transformer ces derniers pour fournir un maillage en quadrilatères 4-nœuds (ce sera l'objet du paragraphe suivant) . A partir de l'une ou l'autre de ces situations, nous pouvons fournir un maillage en triangle 6-nœuds et quadrilatères 8-nœuds respectivement .

La transformation se fait par simple ajout d'un nœud au milieu de chaque arête . Il s'agit donc de toutes évidence d'une transformation en éléments subparamétriques . Cependant, pour être conforme avec notre structure de données, nous projetons les points apparus sur un côté situé sur une arête courbe .

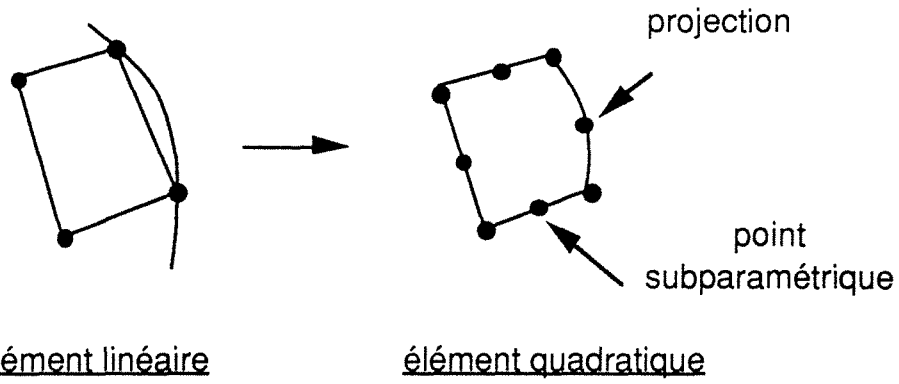


Figure III-28

La réactualisation de l'ensemble de la structure de données est automatique .

III-5-4 :Utilisation de transformations géométriques

Cet utilitaire est semblable à celui développé dans le cadre du mailleur par bloc . Il utilisa la définition de "motifs" réactualisables vue au chapitre II, permettant une utilisation sans risque du processus . Comme pour le mailleur par bloc, il y a une profonde refonte des données, notamment des frontières, pour décrire correctement le nouveau domaine .

Pour terminer, on peut dire que ces utilitaire associé au programme de base, assurent de bonnes performances au logiciel, tant en simplicité, et en souplesse qu'en efficacité et en sûreté .

Dans la mesure du possible, la validité de chaque transformation demandée est testé, puis le processus s'effectue automatiquement . Dans la mesure où ils sont compatibles, ces utilitaires peuvent être appelés dans n'importe que ordre .

III-6 : Transformation en quadrilatères

III-6-1 : Introduction

S'il existe plusieurs techniques de maillage automatique en triangles , par contre, à notre connaissance, il n'existe pas de mailleurs en quadrilatères totalement automatiques . Il faut dire qu'il existe des configurations pour lesquelles il n'y a pas de solution . De plus , dans beaucoup d'autres cas, la solution existante conduit à des éléments de mauvaise

qualité qu'il n'est pas toujours possible de corriger . Bref, le quadrilatères est, géométriquement parlant, un drôle de client .

Une des solutions proposée est de panacher les éléments (généralement triangles et quadrilatères), les triangles étant là pour "boucher les trous" et assurer à moindre frais la conformité géométrique du maillage . Ce type de panachage doit vérifier les conditions de continuité entre éléments de type différent pour être viable [EL 1] [EL 2] .

C'est là une solution assez efficace mais qui peut poser des problèmes au niveau des pré- et post-traitements . Nous avons donc rejeté cette solution .

Nous avons plutôt essayé de transformer la triangulation de Delaunay en maillage quadrilatère par regroupement d'éléments

III-6-2 : Passage de triangle aux quadrilatères

Une des solutions les plus simple serait de diviser chaque triangle en trois, comme indiqué sur la figure III-29 :

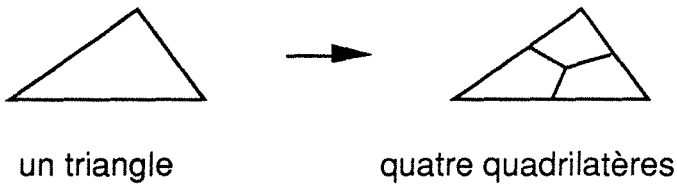


Figure III-29

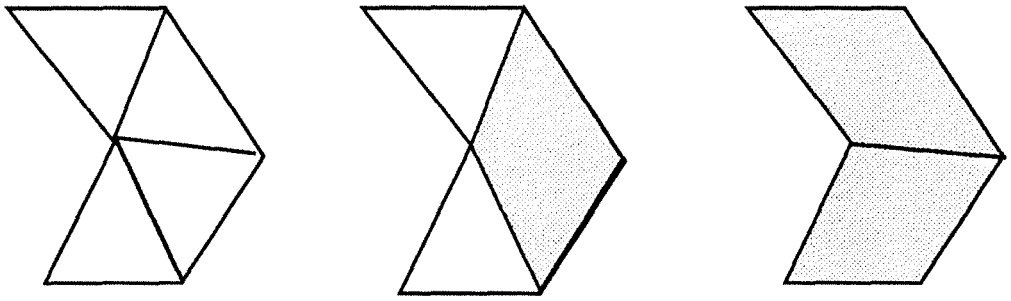
Une seule constatation ; quelle horreur ! . On arrive à transformer un bon maillage en triangle en mauvais maillage quadrilatères (un lissage n'arrange que peu les choses) . En plus, le nombre d'éléments est divisé par trois ... Bref, nous le rejetons d'emblée comme seul processus de transformation . Une fois adapté, ce processus permettra de nous tirer d'embarras, comme nous le verrons plus loin .

Le processus que nous utiliserons effectivement est celui du regroupement de triangles 2 à 2 pour former un quadrilatère. Les quadrilatères 4-nœuds étant des éléments sensiblement plus riches que les triangles, nous nous inscrivons dans un processus de diminution du nombre d'éléments compensé par une augmentation de leur performances . Nous procédons comme suit (figure III-30) .

situation de départ résultat

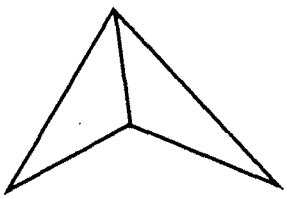
Figure III-30

De toute évidence, une telle transformation n'est pas toujours possible . De plus, il est des regroupements qui sont plus judicieux que d'autres, remarques illustrés par la figure III-31 :



situation de départ résultat (a) résultat (b)

Le résultat (b) résoud complètement le problème alors que (a) laisse deux triangles inutilisés



regroupement impossible

Figure III-31

Nous avons donc décidé de regrouper les éléments selon les critères suivants de priorité:

1°- Regrouper les éléments pour lesquels il n'y a pas d'autre possibilité .

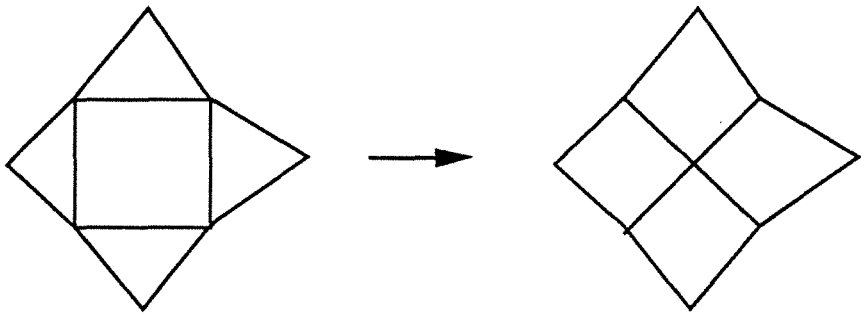
2°- Si l'on a le choix, choisir le regroupement offrant la meilleure qualité de quadrilatères .

On notera qu'aucun point nouveau n'apparait, il s'agit donc de maillages équivalents .

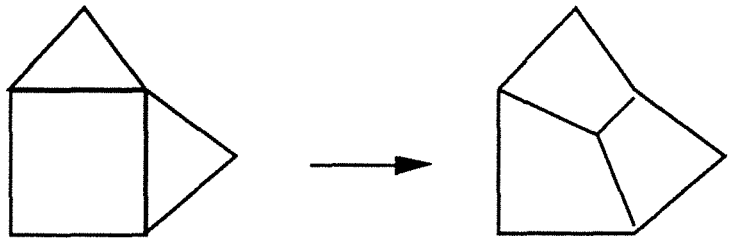
Malheureusement, même avec ces précautions, il se peut que des triangles subsistent.

III-6-3 : Post traitement

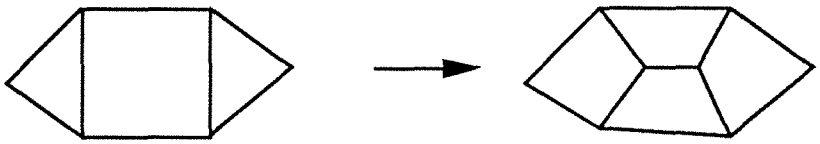
Nous proposons un processus de regroupement quadrilatères-triangles. On va s'efforcer de trouver un maillage localement équivalent à des groupes quadrilatères-triangles sans avoir besoin de modifier le reste du maillage . Voici quelques exemples :



Un quadrilatère et 4 voisins triangulaires



Un quadrilatère et 2 voisins triangulaires adjacents



Un quadrilatère et 2 voisins triangulaires opposés

Figure III-32 ; regroupement d'éléments

Les points intermédiaires crées sont pris au barycentre de certains sommets choisis judicieusement .

Cependant, le cas "1 quadrilatère et un voisin triangulaire" n'a pas de solution simple (c'est à dire de solution locale ne nécessitant pas un remaillage total) . Le cas de "3 voisins triangulaires" peut se ramener au cas précédent par regroupement du quadrilatère et de deux triangles

Nous constatons donc qu'en dépit de nos efforts, on ne peut assurer la totale disparition des éléments triangulaires . Pour pallier à cet inconvénient, nous choisissons de diviser de manière homogène le maillage . Les quadrilatères sont divisés en 4, les quelques triangles restant en 3 selon la figure III-33 .

division finale du maillage en quadrilatères

Figure III-33

Après division, l'ensemble du maillage est lissé par un processus analogue à celui défini dans le paragraphe 5 . La convergence étant plus lente, nous effectuons un peu plus d'itérations que dans le cas des triangles . Ce lissage est absolument indispensable quand on voit la mauvaise qualité des éléments crées .

III-6-4 : Conclusions

Bien que le maillage ne quadrilatère par cette sous-méthode soit correct au point de vue des éléments finis, son équilibre est cependant assez mauvais . L'automaticité est au prix de la performance .

En matière de maillage en quadrilatères, il semble que cette idée de regroupement ait été abandonnée (vu le rapport efficacité/complexité, on le comprend) au profit de maillage par propagation des bords jusqu'à l'intérieur, semblable à celui existant pour les triangles ([EL 8], mailleur TRIGEO) . Si l'algorithme semble converger, nous n'avons pas eu connaissance des résultats .

III-8 ; Organigramme du programme

Le déroulement complet du programme peut être schématisé sous la forme suivante :

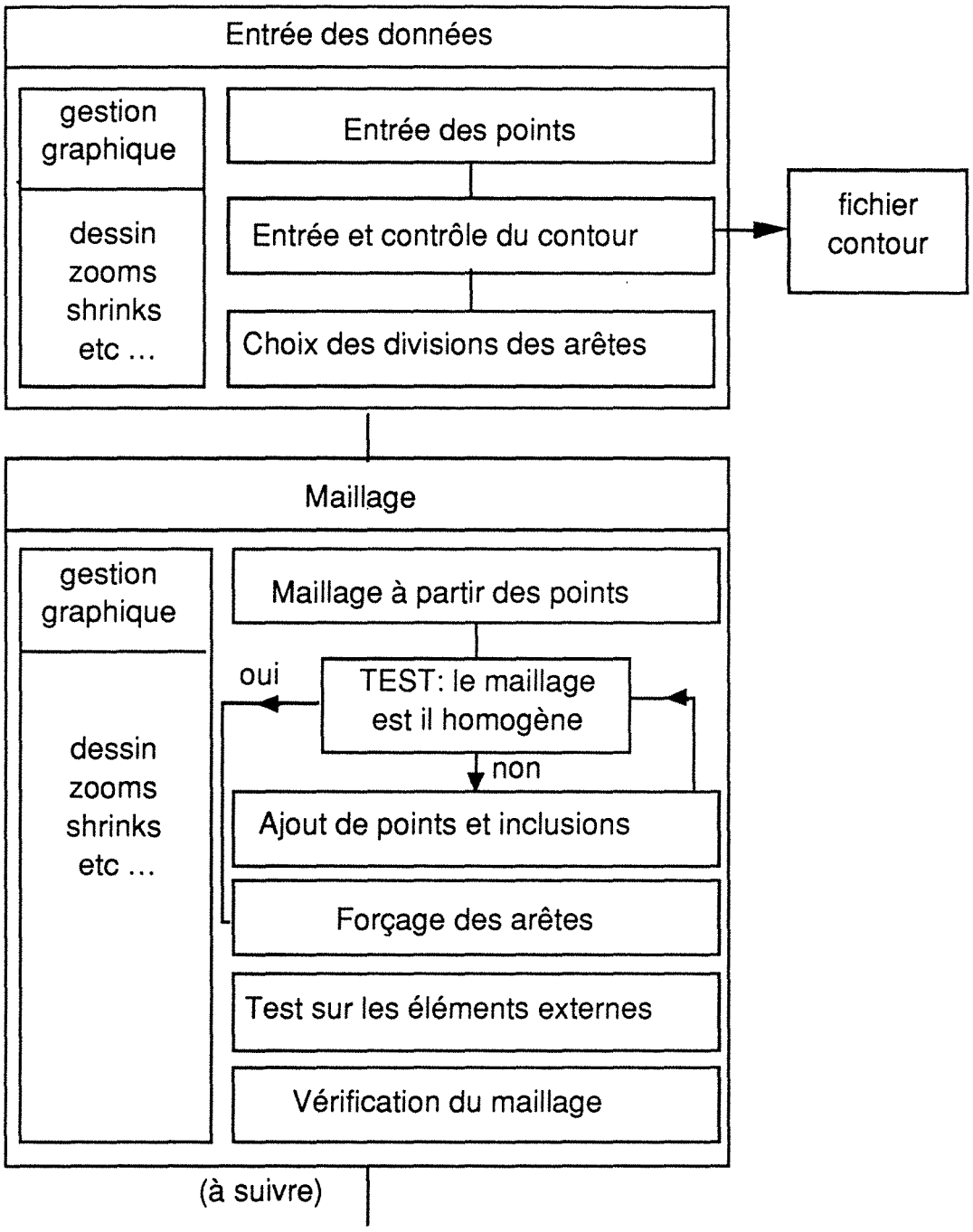


Figure III-34 ; début

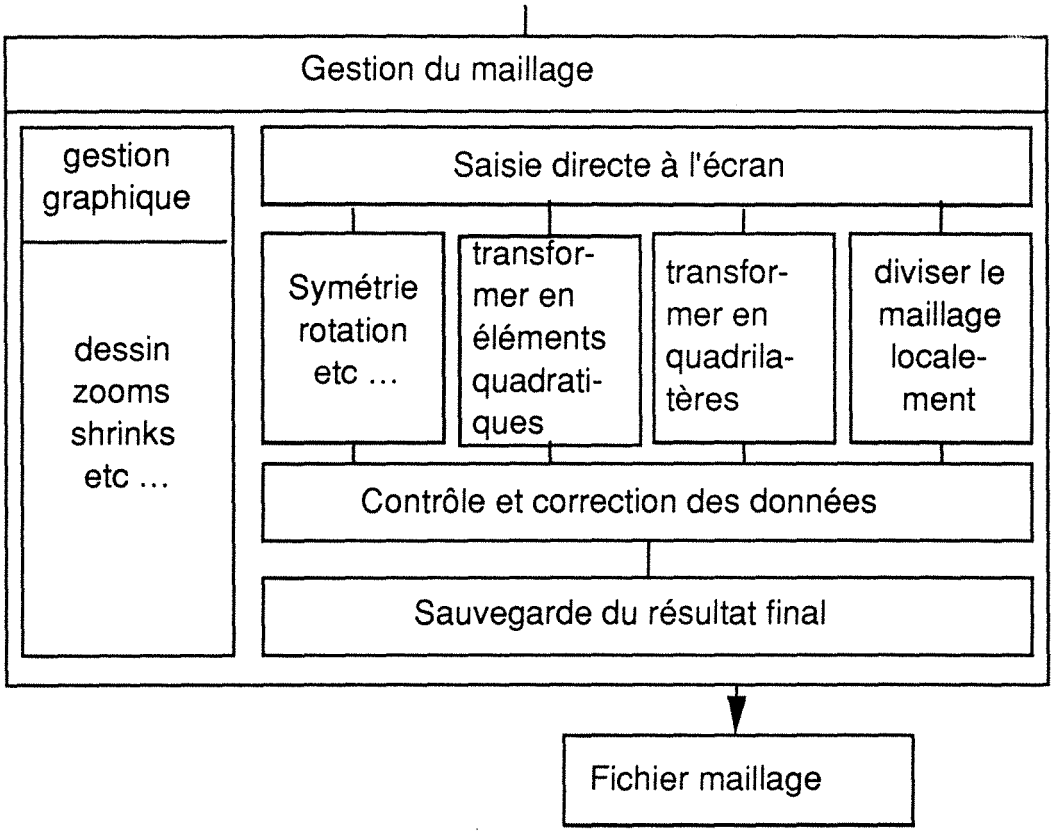


Figure III-34 ; organigramme complet

III-8 : Exemples

Nous présentons maintenat quelques exemples de réalisation illustrant les possibilités du mailleur .

III-8-1 : Rotor de moteur

Cet exemple pratiquement identique à celui vu au chapitre II illustre les deux méthodes fondamentales de maillage :normale (figures indicées (a)) ou modifiée (figures indicées (b)) . On constatera que les résultats produits sont identiques . Il permet aussi d'illustrer les divers utilitaires géométriques permettant d'augmenter l'efficacité du maillage.

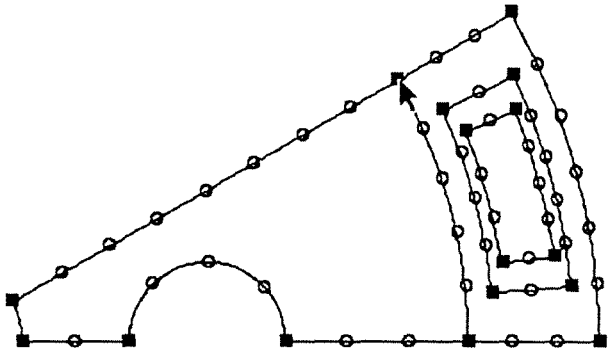


Fig III-35 : Contour et points limites
Le point de départ du maillage (58 points)

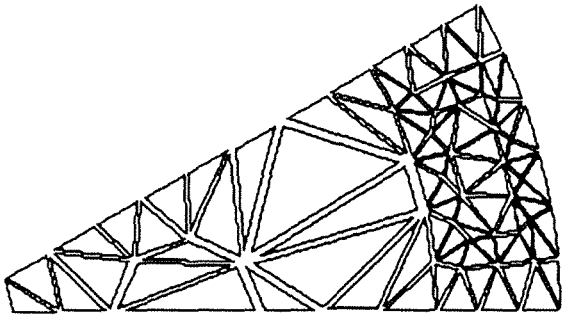


fig III-36-a

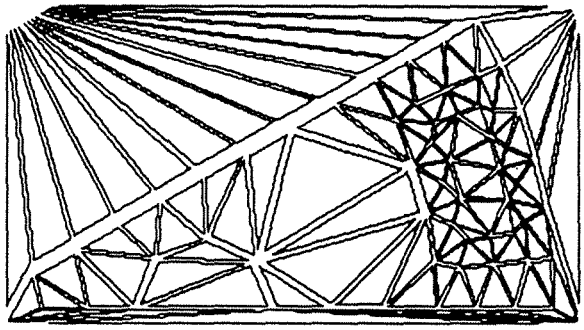


fig III-36-b

Le maillage à partir de la peau . On remarquera la disparité des éléments et le non-respect du contour (au niveau du double rectangle central).

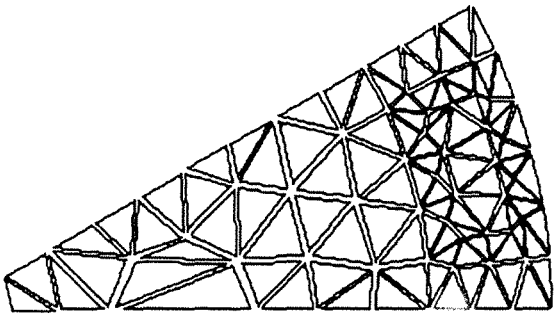


fig III-37-a

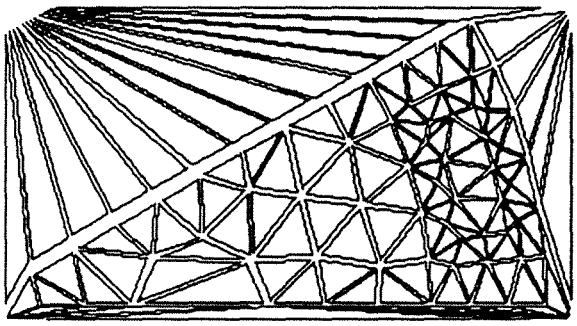


fig III-37-b

Dans les deux cas, le maillage est plus régulier, mais il ne respecte toujours pas son contour .

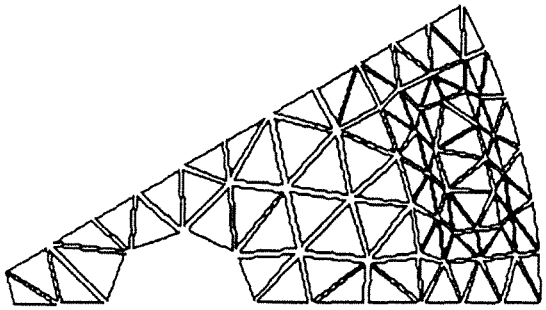


fig III-38-a

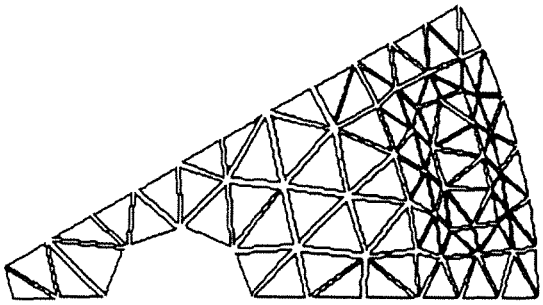


fig III-38-b

Après forçage des arêtes, dans les deux cas, le maillage respecte son contour . Les éléments extérieurs ont été éliminés . On constate la similitude des maillages .

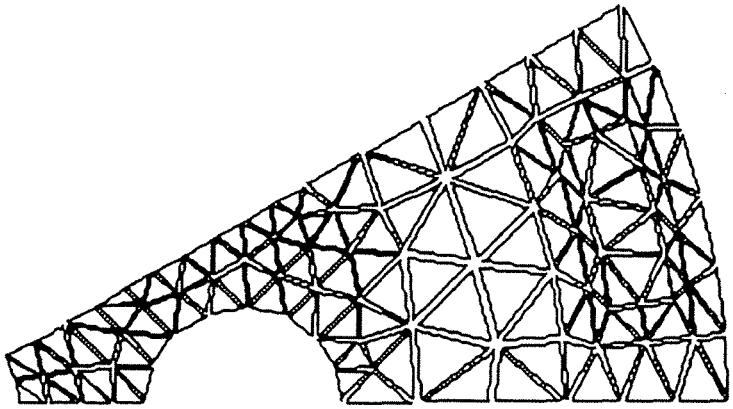


Figure III-39 : Division locale de la pointe
(94 points, 144 éléments) . On notera les projections sur les frontières courbes .

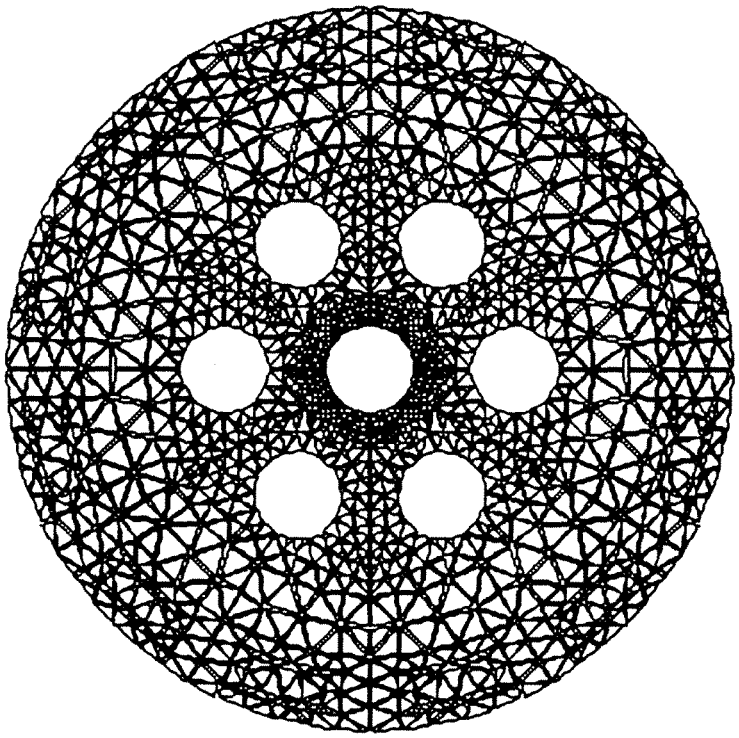


Figure III-40 : maillage total (6 symétries)

(1124 points, 2052 éléments) . Temps total de calcul sur Mac II (tout compris) : 30 minutes.

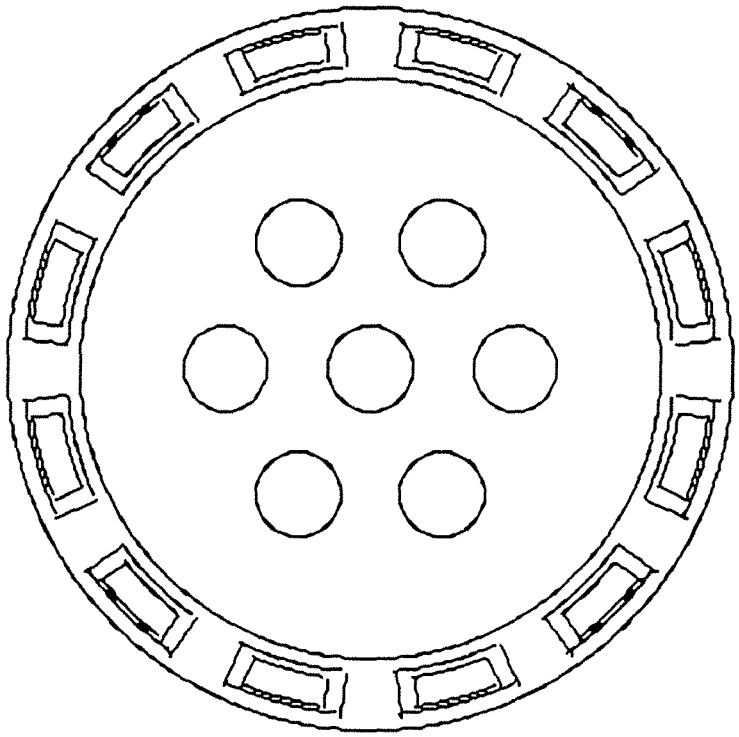


Figure III-41 : Le contour réactualisé avec suppression de nombreuses arêtes internes .

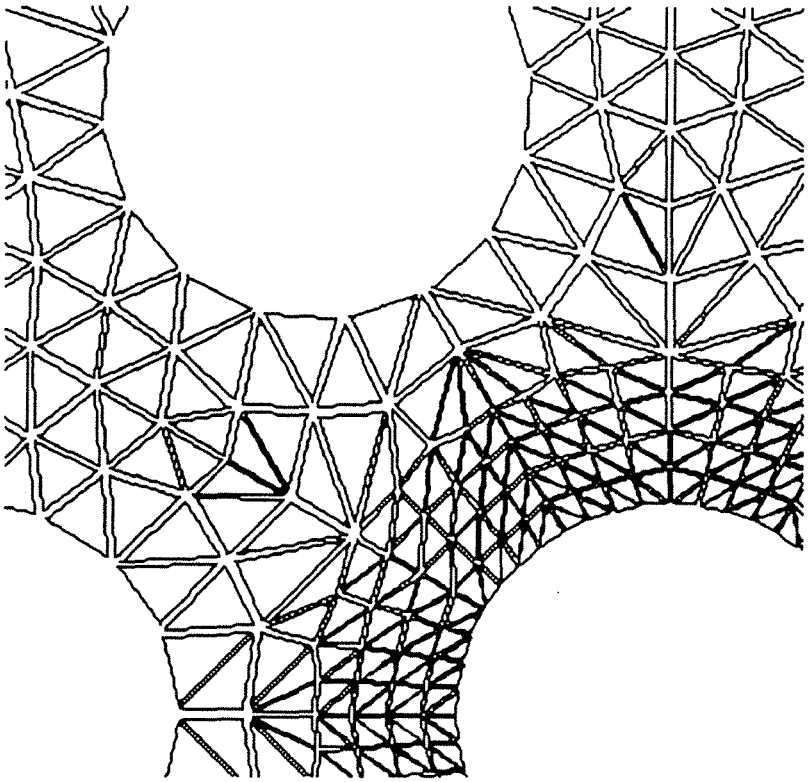


Figure III-42 : Zoom du trou central

III-8-2 : Domaine fortement non convexe

Nous présentons un exemple de domaine fortement non-convexe : Il s'agit d'une tête de piston de moteur Diesel .

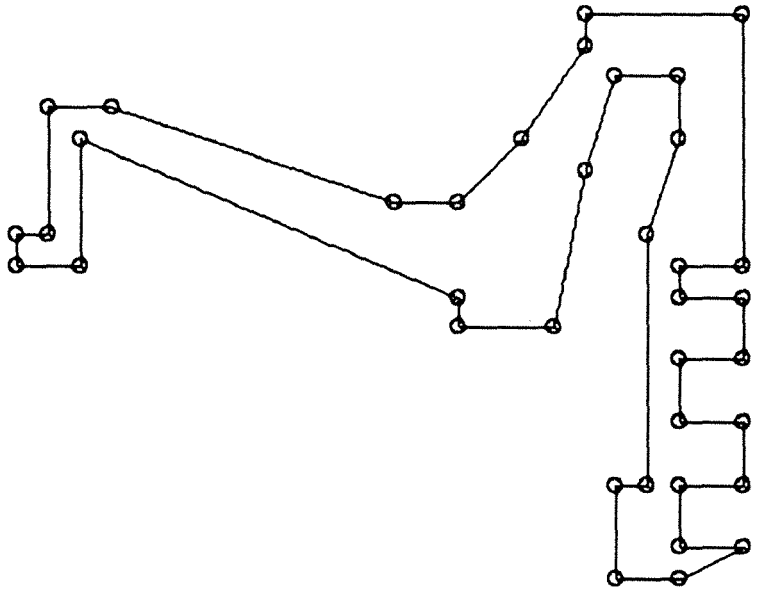


Figure III-43 : Contour du domaine

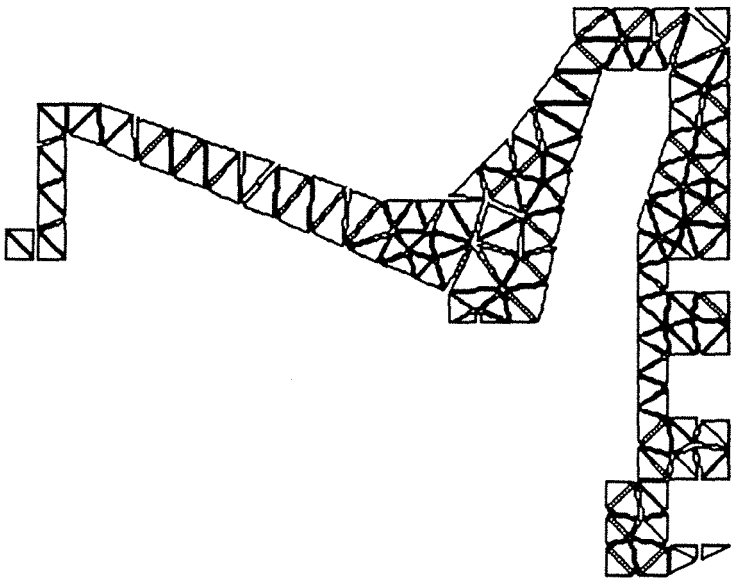


Figure III-44 : Maillage réalisé
(134 points, 162 éléments)

III-8-3 : Domaine avec trous

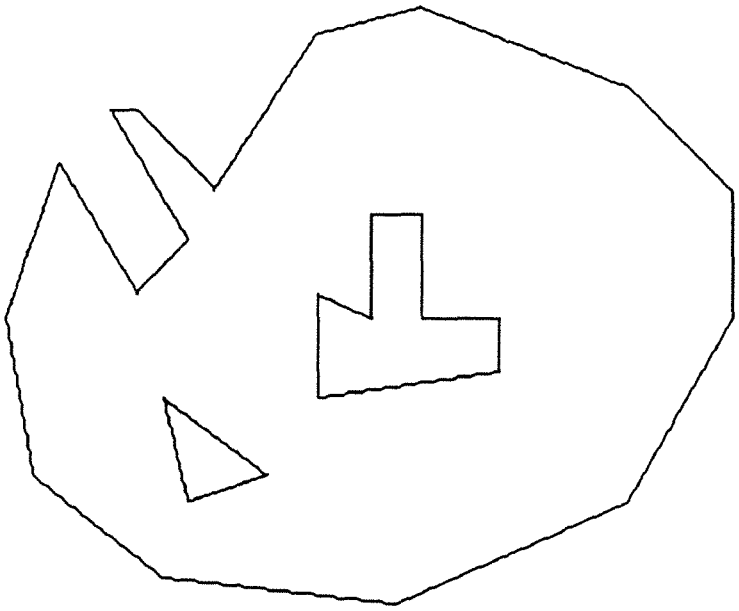


Figure III-45
Le contour de départ contenant deux trous et composé de 3 contours et de 27 arêtes droites.

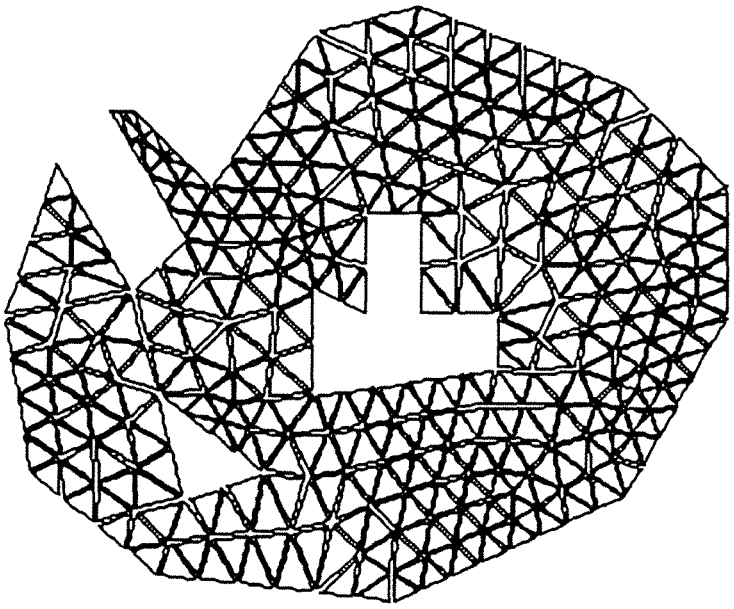


Figure III-46
Le maiilage final (236 points , 436 éléments, 98 points-frontières)

III-8-4 : Passage aux quadrilatères

Nous reprenons l'exemple précédent pour transformer la triangulation en maillage de quadrilatères . Comme on pourra le constater, le résultat est plutôt moyen .

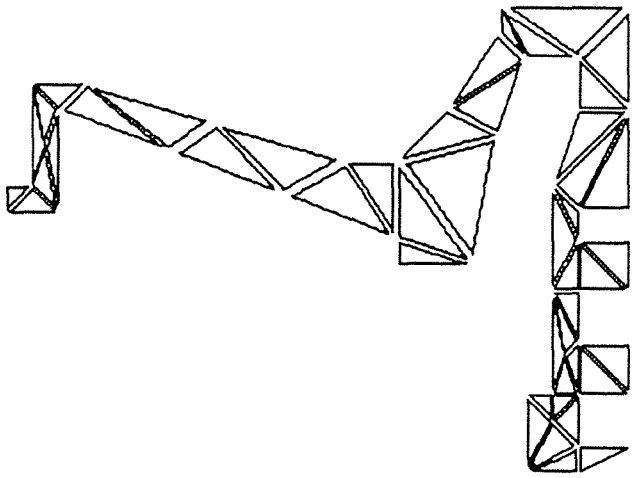


Figure III-47 : maillage en triangles
(45 points, 46 éléments)

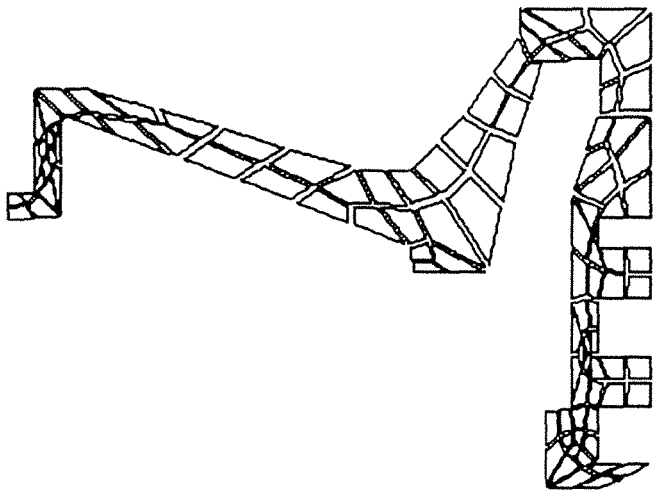


Figure III-48 : maillage en quadrilatères
(146 points, 104 éléments)

III-9 ; Conclusions

A l'exception peut être de certains aspects du maillage en quadrilatères, la méthode de Delaunay fournit un outil simple, fiable et souple d'emploi qui nous a donné toute satisfaction . Comme pour l'ensemble des autres programmes, un effort particulier a été fait sur la présentation, l'accessibilité des utilitaires, l'automatisme des transformations et le contrôle et la sécurité des fonctions .

Le logiciel représente un ensemble de près de 8000 lignes; il a été testé avec succès sur des maillages variés et complexes de plus de 2000 triangles sans provoquer d'erreur.

Chapitre IV

Maillages en dimension 3

IV-1 : Introduction

La nécessité du recours à des mailleurs automatiques est encore plus évidente en dimension 3 qu'en dimension 2. D'abord parce qu'une dimension supplémentaire entraîne un fort accroissement de la quantité d'informations à stocker par rapport à celle exigée en dimension 2 (très grossièrement, et à densité égale, environ le carré de la quantité en 2D). Ensuite, parce que si le problème tridimensionnel se formule géométriquement comme le problème bidimensionnel, de façon pratique, les informations à fournir, notamment concernant le contour, sont beaucoup plus nombreuses et peuvent dépendre de la méthode de maillage utilisé.

Nous n'avons pas la prétention de présenter ici des algorithmes complets, mais plutôt quelques utilitaires de maillage 3D fournissant cependant des résultats tout à fait acceptables dans leurs limites d'utilisation. Ces utilitaires sont le prolongement d'études bidimensionnelles ainsi que le fruit de certaines réflexions sur les méthodes multigrilles en dimension 3.

Nous présenterons d'abord une ébauche de logiciel convivial utilisant essentiellement un contrôle graphique du maillage. Dans cette optique, nous présenterons un utilitaire général de calcul des parties cachées indispensable à ce genre de logiciel, mais ô combien coûteux en temps de calcul.

Nous terminerons par quelques idées de développements futurs qui devraient certainement être utilisés avec profit dans les prochains algorithmes de maillage 3D.

IV-2 : Généralités sur la représentation de pièces 3D

IV-2-1 : Représentation en perspective

La représentation d'une figure tridimensionnelle ne peut se faire sur un écran que par l'intermédiaire d'une projection en perspective. Dans cette optique, il est d'abord nécessaire de se fixer quelque paramètres.

Soit une pièce F, connue par un ensemble de données (points ou faces par exemples) exprimées dans un repère de référence Ro $(o,\vec{x},\vec{y},\vec{z})$. Ce repère permet de définir F de façon absolue.

Pour pouvoir définir une perspective, il faut se donner un "point de vue" d'où l'on observe la figure F. Ce point que nous désignerons par O', est connu par ses coordonnées exprimées dans le repère R_O. Dans la pratique, on exprime souvent la position de O' par des coordonnées sphériques centrées sur O, ce qui réduit la connaissance de O' à trois valeurs θ, ϕ et r, conformément à la figure IV-1

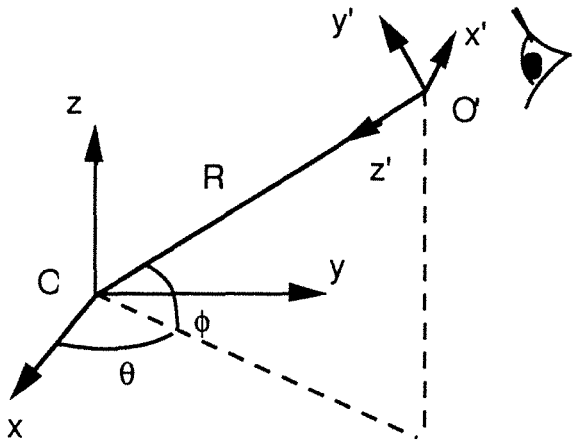


Figure IV-1: repère de référence et repère observateur

Sur le point O', nous construisons un repère R' $(o',\vec{x}',\vec{y}',\vec{z}')$ appelé repère de l'observateur, construit comme indiqué sur le schéma ci-dessus (notons que le repère R' est indirect).

Les coordonnées de la pièce F exprimées dans le repère R' permettent de déterminer la façon dont cette pièce est vue par l'observateur situé en O'. Le passage des coordonnées d'un repère à l'autre est donné par les formules suivantes :

Un point M de coordonnées (x,y,z) dans R_0 est transformé en un point de coordonnées (x',y',z') dans r' telles que :

$$\begin{aligned} x' &= -x.\sin\theta + y.\cos\theta \\ y' &= -x.\cos\theta.\sin\phi - y.\sin\theta.\sin\phi + z.\cos\phi \\ z' &= -x.\cos\theta.\cos\phi - y.\sin\theta.\cos\phi + z.\sin\phi + R \end{aligned} \qquad (1)$$

Pour une représentation plane, il faut maintenant se donner un plan de projection P, perpendiculaire à l'axe O' z' et connu par sa distance D par rapport au point O'. Les coordonnées de la projection d'un point M' de coordonnées (x',y',z') dans R' sont alors dans le repère du plan P (x", y") telles que :

$$\begin{aligned} x'' &= \frac{D.x'}{z'} \\ y'' &= \frac{D.y'}{z'} \end{aligned} \qquad (2)$$

Dans le cas d'une projection à l'infini, ces formules se limitent à :

$$\begin{aligned} x'' &= x' \\ y'' &= y' \end{aligned} \qquad (2')$$

Soient les figures ci-dessous (figure IV-2)

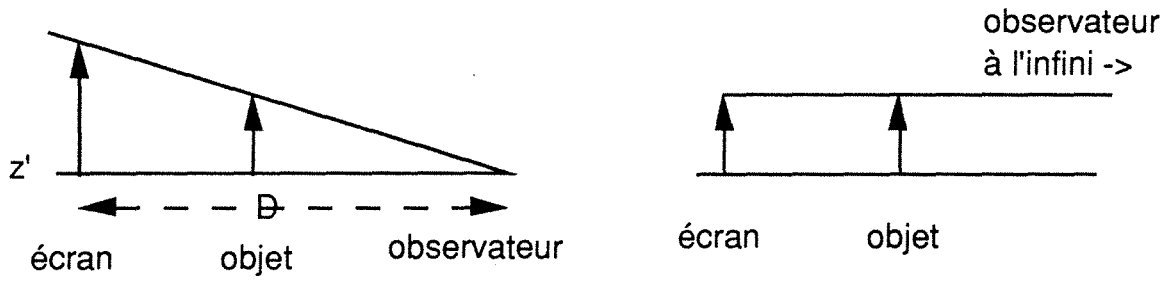


Figure IV-2 : types de projections

On notera dans toutes ces définitions, l'importance de l'axe z'. Il constitue " l'axe de profondeur" du dessin; plus un point de coordonnées (x',y',z') est tel que z' est grand, plus ce point est loin de l'observateur.

Voyons maintenant l'influence de paramètres de vision. Le rôle de θ et ϕ est simple : faire tourner la pièce dans tous les sens pour obtenir un point de vue différent. Il n'ont aucun rôle dans la déformation de l'objet. Il en va autrement pour R. Plus R est petit, plus l'image est déformée par la perspective (Figure IV-3).

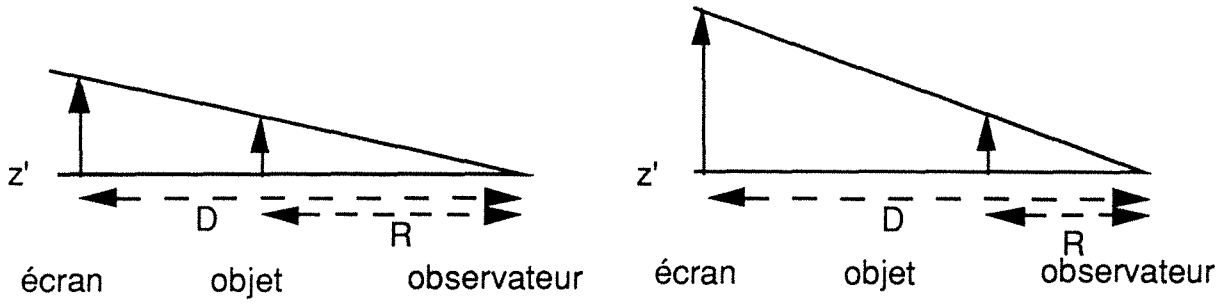


Figure IV-3 : Influence de R

Le paramètre D, lui, fait varier la taille de l'objet, mais il n'engendre pas de déformation autre que celle existant déjà due à R. La taille de l'image varie dans le même sens que D.

Pour éliminer ces déformations, il faut donc augmenter R et compenser la diminution résultante de l'image par une augmentation de D.

Nous avons donc en résumé quatre types de coordonnées :

- Les coordonnées de référence de la pièce sur lesquelles on travaille rarement.

- les coordonnées de l'observateur définissent la pièce telle qu'on la voit. Ce sont nos coordonnés de travail.

-Les coordonnées de projection décrivant la perspective.

- Enfin, les coordonnées entières de l'écran. il s'agit de coordonnées discrètes, et directement liées aux précédentes Leur définition dépend de la taille de l'écran graphique de l'ordinateur.

Les deux premiers systèmes de coordonnées définissent un même espace de dimension 3 , appelé espace objet. Ils permettent de caractériser la pièce sur laquelle on travaille. Notons que le premier repère est direct et que le second est indirect.

Les deux derniers repères définissent l'espace image. La transformation faisant passer de l'espace objet à l'espace image n'est pas bijective (deux points différents de l'espace objet peuvent avoir la même image). De plus, les coordonnées entières introduisent des erreurs d'arrondi lors de leur calcul, ce qui fait que deux points de l'espace image proches peuvent se retrouver confondus. Enfin, si l'on prévoit de faire des zooms, il sera nécessaire de recalculer toutes les coordonnées image entières sous peine d'obtenir des résultats aberrants.

Si l'on souhaite effectuer des calculs précis, quasi-analytiques (à la précision de l'ordinateur), le travail dans l'espace objet est indispensable. Cependant, ces calculs sont très lents et complexes (selon le mode de définition des entités géométriques intervenant. Il est possible dans d'autres cas (certains algorithmes de calcul des parties cachées, de rendu réaliste etc ...) de travailler dans l'espace image, et notamment en coordonnées entières. Ceci a l'avantage d'offrir moins de calcul et donc d'être plus rapide , en contrepartie, cela présente l'inconvénient d'être plus une trace fugace à l'écran qu'une réalité géométrique. De plus, tout est à recalculer lors d'un zoom.

On choisira donc l'un ou l'autre des espaces selon le but recherché (précision ou rapidité) et les impératifs imposés.

IV-2-2 : Coordonnées homogènes

IV-2-2-a : Définitions

L'utilisation des coordonnées homogènes est basée sur le principe des fraction : on sait que $\frac{a}{b}$ et $\frac{p.a}{p.b}$ $(a \neq 0$ et $p \neq 0)$ représentent le même nombre .

L'idée est donc d'augmenter d'un le nombre des coordonnées d'un point p=(x,y,z) pour obtenir p'= (qx,qy,qz,q) (q≠0). Ainsi , pour r≠ 0 et s≠0 , (rx,ry,rz,r) et (sx,sy,sz,s) représentent le même point.

Le passage des coordonnées normales aux coordonnées homogènes est simple : il suffit de prendre r=1 :

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} \Rightarrow \begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix} \quad (3)$$

L'inverse est tout aussi simple :

$$\begin{Bmatrix} x \\ y \\ z \\ t \end{Bmatrix} \Rightarrow \begin{Bmatrix} x/t \\ y/t \\ z/t \end{Bmatrix} \quad (t \neq o) \quad (4)$$

Nous reparlerons plus loin des avantages de cette écriture, mais nous pouvons quand même faire état de deux points importants.

- la possibilité de traiter toute transformation infographique de base sous forme matricielle. Ceci est particulièrement rentable pour les projection et les symétries.
- la limitation du nombre de divisions numériques.

IV-2-2-b : Propriétés

Si nous considérons une application linéaire qui , en coordonnées non-homogènes, fait passer: $\vec{P} : (x,y,z)$ à $\vec{p'} : (x',y',z')$ tel que :

$$x'_i = \sum_{j=1}^{3} a_{ij} x_j \quad \text{soit} \quad \vec{P'} = A.\vec{P} \quad \text{avec A matrice } [3x3]$$

Alors, il est possible de composer une matrice [4x4] A valant

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5) \quad \text{telles qu'on ait :} \begin{Bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ 1 \end{Bmatrix} = [A] \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{Bmatrix} \quad (5')$$

La matrice en coordonnées homogènes de la composée AoB de deux applications linéaires sera le produit matriciel [A] x [B], il n'y a là rien de surprenant.

L'intérêt de la méthode provient du fait qu'il est possible de mettre sous forme matricielle des transformations qui ne sont pas des applications linéaires (notamment les translations ou les projections). Donnons par exemple la matrice d'une translation en
coordonnées homogènes de vecteur $\vec{V} = (a,b,c)$.

$$\begin{Bmatrix} x' \\ y' \\ z' \\ 1 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix} \qquad (6)$$

De façon générale, il est facile d'exprimer les matrices de transformations simples (transformation par rapport aux plans ou aux axes de coordonnées).

On peut alors en tirer deux conclusions importantes :

1) _ Il devient facile d'expliciter la matrice [A] d'une transformation quelconque. Il suffit pour cela de décomposer la transformation en une composée de transformations simples dont on sait calculer les matrices (on peut par exemple ramener par une suite de rotations et de translations, une symétrie de plan quelconque en une symétrie par rapport à un plan de coordonnées).

2) _ De façon plus générale, une composée de transformations quelconque n'est finalement plus représentée que par une matrice [4x4]. Le traitement d'un grand nombre de points est donc obtenu sans grande difficulté et de façon automatique.

Nous utilisons intensivement les coordonnées homogènes, ce qui confère au logiciel une plus grande souplesse ainsi qu'une plus grande rapidité.

IV-3 : Mailleur par blocs

II-3-1 : Principe

Cet algorithme développe une méthode analogue à celle utilisée dans le mailleur par bloc bidimensionnel. Cette fois, l'idée de base consiste à ramener l'élément réel considéré à un élément idéal centré sur [0,0,0] et de côté égal à 2. Dans cet élément que nous appelerons élément de référence, nous pouvons effectuer un maillage de base en hexaèdres à partir d'un nuage de point construit couche après couche, comme le montre la figure IV-4.

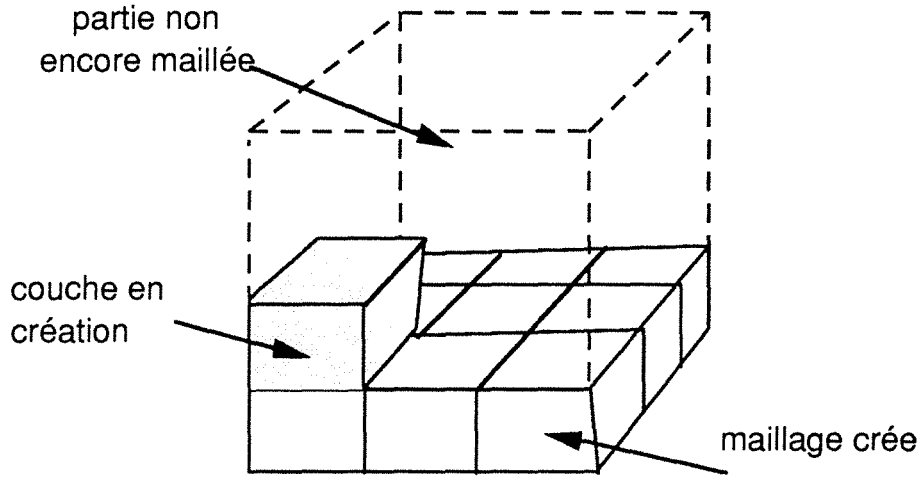


Figure IV-4 : maillage par couche de l'élément de référence

Parlons maintenant de l'élément réel. Nous avons choisi comme élément réel, l'élément isoparamétrique 20-nœuds [EL 2] figure IV-5.

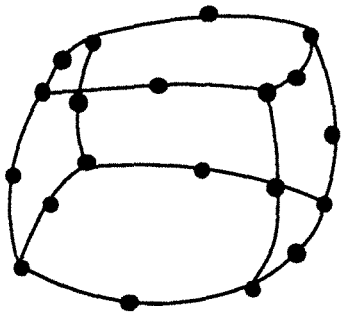


Figure IV-5 : élément réel 20-nœuds

Il s'agit d'un élément cubique à faces courbes. Pour passer de l'élément de référence à l'élément réel, nous utilisons les fonctions de forme N_i de cet élément. Rappelons que ces 20 fonctions (une par sommet ou nœud milieu définissant l'élément) vérifient la propriété suivante:

$$N_i(\xi_j,\eta_j,\zeta_j)=\delta_{ij} \quad (7)$$

où (ξ_j, η_j, ζ_j) sont les coordonnées dans le repère de référence des 20 points définissant l'élément de référence.

On passe d'un point de coordonnées (ξ, η, ζ) dans le repère de référence à un point de coordonnées (x,y,z) dans le repère réel par les relations :

$$\begin{aligned} x &= \sum_{i=1}^{20} N_i(\xi,\eta,\zeta)\, x_i \\ y &= \sum_{i=1}^{20} N_i(\xi,\eta,\zeta)\, y_i \quad (8) \\ z &= \sum_{i=1}^{20} N_i(\xi,\eta,\zeta)\, z_i \end{aligned}$$

Où (x_i,y_i,z_i) sont les coordonnées définissant l'élément réel.

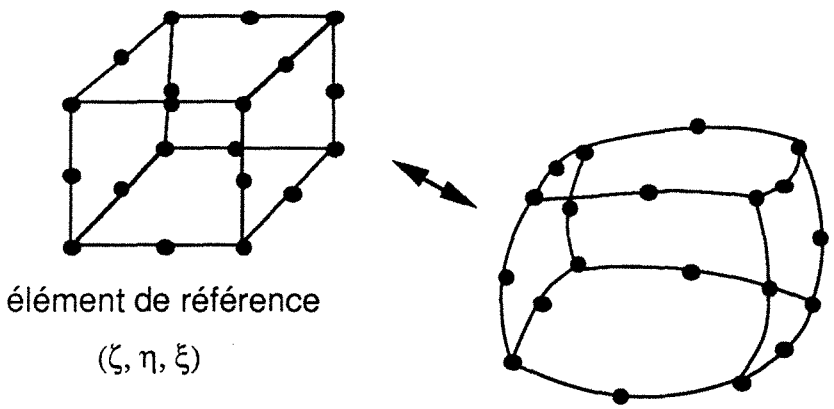


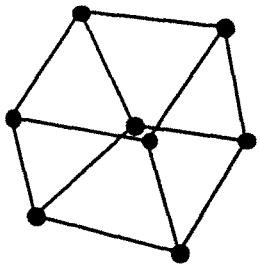
Figure IV-6 : élément de référence et élément réel

Par analogie avec le cas bidimensionnel, nous construisons des éléments du type hexaèdres (à bords droits ou courbes) en nous appuyant sur le réseau des trois nappes isoparamétriques définies par les équations ξ=constante, η=constante, ζ=constante. Ces trois réseaux de nappes sont orthogonales entre elles, ce qui assure une qualité d'éléments aussi bonne que possible.

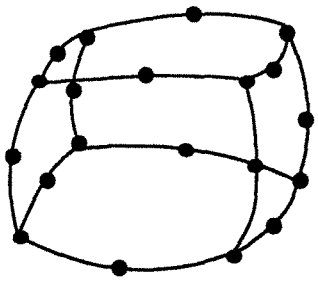
D'un point de vue analytique, il faut que la transformation $(x,y,z) = \mathbf{F}(\xi,\eta,\zeta)$ soit bijective pour assurer un maillage correct. Contrairement au cas bidimensionnel, la conformité de la transformation ne peut être assurée par quelques critères géométriques simples. Il faut un contrôle numérique constant du signe du Jacobien de la transformation. Une telle vérification est lourde à mettre en œuvre et pénalise le temps de calcul. Nous nous limiterons donc au fait que l'image par la transformation des faces de l'élément de référence donnent les faces de l'élément réel. Ce type de contrôle pourra être aidé par une visualisation poussée de la structure de données entrée.

II-3-2 : Principe général de maillage par blocs

Comme nous venons de le voir, le processus cité permet d'obtenir des éléments de forme hexaédrique. A partir de ces données, nous limiterons notre mailleur à la construction des éléments suivants (figure IV-7):



quadrilatère 8 nœuds



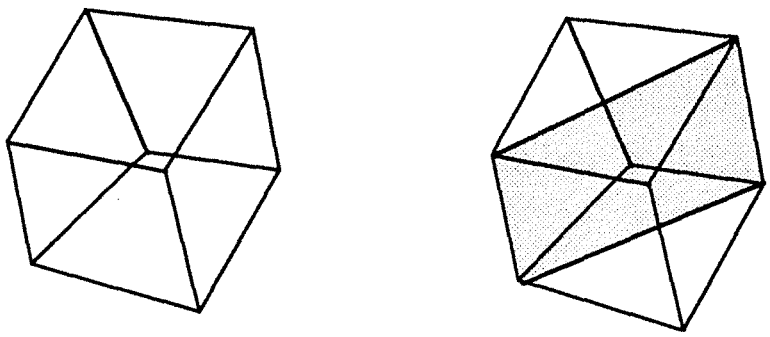
quadrilatère 20 nœuds

Figure IV-7 : divers type d'éléments obtenus

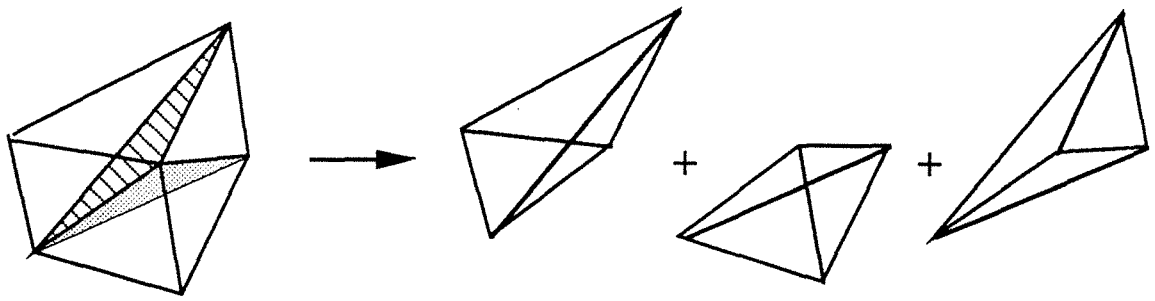
Chaque bloc est maillé indépendamment des autres. Le résultat est mémorisé par le stockage des tétraèdres (8-nœuds ou 20-nœuds) et des points crées. Chaque hexaèdre est décrit par l'ensemble de ses sommets parcourus selon un ordre fixé.

Bien entendu, se repose ici le problème des points de coordonnées identiques appartenant à des blocs différents. Nous résolvons le problème par la fusion automatique de ces nœuds communs, ce qui conduit à une correction dans les connections des éléments.

Enfin, signalons la possibilité de décomposer le maillage obtenu dans le cas des hexaèdres 8-nœuds en tétraèdres 4-nœuds par simple division selon le schéma suivant :



Phase 1: décomposition en deux pentaèdres par coupure suivant une bissectrice



Phase 2: décomposition en 3 tetraèdres

Figure IV-8 : division en tétraèdres

A ce stade, nous pouvons utilement résumer notre processus de maillage par l'algorithme suivant :

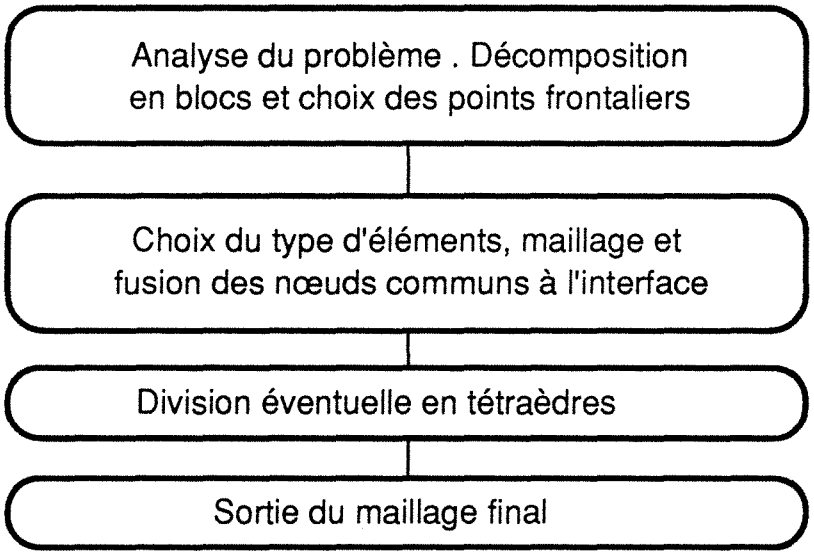


Figure IV-9 : algorithme de maillage par blocs

IV-3-3 : Structure générale de données

Comme en dimension 2, on commence par décomposer le domaine en blocs (à 20 nœuds). Il s'agit là d'un travail pénible et minutieux. Pour alléger cette tâche, nous avons mis au point plusieurs utilitaires aidant l'utilisateur. Tout bloc doit être décrit par la donnée de ses 20 nœuds; pour ce faire, nous proposons les facilités suivantes:

_ entre des points en coordonnées cartésiennes, cylindriques ou sphériques au choix.

_ la possibilité de déduire certains points de points existant déjà par des transformations géométriques simples (symétries, rotation etc ...).

_ L'utilisation d'une petite bibliothèque de formes pré-établies (cubes, trapèzes, cylindres...).

_ La possibilité de reproduire certain blocs par des transformations simples (symétries, rotation etc ...).

Les frontières du domaine sont automatiquement déduites de la donnée des blocs. Chaque bloc possède 6 faces composées chacune de 8 sommets. Lors de la décomposition du domaine, toute face intégralement commune à deux bloc est considérée comme interne et est donc supprimée du contour. Le contour du domaine se décompose finalement en une série de faces gauches (composée chacune de 8 sommets). Cet ensemble de faces permet de remplir deux tâches distinctes :

1 _ la description géométrique du contour du domaine, utile pour tout opération de remaillage, opération courante lors d'un processus multigrilles comme nous le verrons plus loin.

2 _ la saisie de toutes les conditions limites ou efforts imposés sur la frontière.

La structure de données peut se résumer par le schéma suivant :

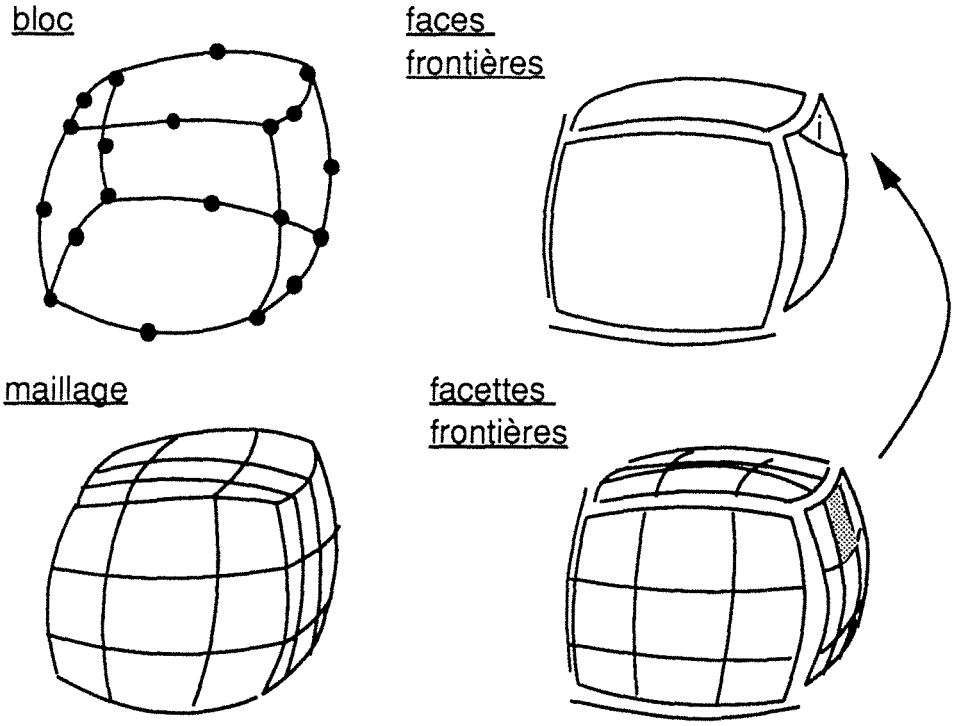


Figure IV-10 : classification des informations

Nous appelerons "facette frontière" toute faces d'un élément du maillage se trouvant sur une face frontière. Cette définition est l'analogue ne dimension 3 de celle reliant les côtés frontières aux arêtes frontières en dimension 2.

La dernière étape avant le maillage consiste à choisir le nombre de nœuds frontaliers suivant les 3 directions de chaque bloc. Nous offrons ici deux possibilités:

_ une répartition linéaire de points le long de l'arête.
_ Une répartition semi-logarithmique centre sur l'une ou l'autre des extrémités de l'arête.

Il ne reste plus à l'utilisateur qu'à choisir le type d'éléments avec lequel il souhaite construire son maillage et à lancer le calcul.

Il est possible de stocker la décomposition du domaine en blocs sur un fichier disque , permettant ainsi une réutilisation du problème. Les informations concernant le domaine sont conservées selon le modèle du chapitre I. Elles comprennent:

_ les coordonnées de chaque point définissant les blocs.
_ la définition de chaque bloc par ses connections
_ L'ensemble des faces frontières.

Le maillage comprend quant à lui

_ les coordonnées de chaque nœud
_ la définition de chaque élément par ses connections
_ L'ensemble des facettes frontières faisant référence aux faces des blocs.

Pour terminer, il est à rappeler que l'ensemble du programme est géré par menus déroulants utilisant autant que possible des saisies directes à l'écran. Cela permet à tout moment une visualisation et un contrôle des résultats aisé. Cependant, la saisie directe à l'écran n'est pas toujours possible car il n'y a plus unicité des coordonnées d'un point. Il n'a donc pas été possible de développer autant d'utilitaires qu'en dimension 2.

IV-4 : Mailleur par topologie cylindrique

IV-4-1 : Principe

Il est courant dans la pratique des éléments finis de devoir mailler des domaines cylindriques. Nous présentons ici un algorithme de maillage construit un maillage tridimensionnel à partir d'un maillage bidimensionnel (dit "couche de base") et d'une transformation donnée déterminant un chemin dans l'espace. Ce maillage est constitué d'un empilement de couches de topologie identique à celle de la couche de base et déduites via la transformation donnée (figure IV-11) [IN 6] [IN 8] [IN 9].

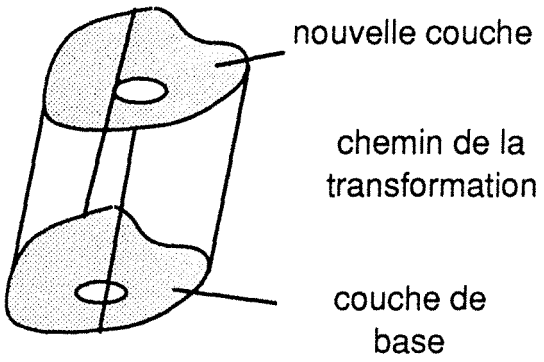


Figure IV-11 : position du problème

On le voit, l'algorithme général n'est pas très compliqué mais il faut être précis vis à vis de certaines difficultés de maillage ainsi que dans la transmission des données, comme nous le verrons par la suite.

IV-4-2 : Structure de données

Nous allons maintenant revenir en détail successivement sur le maillage bidimensionnel, le type de transformation, la construction des éléments et enfin le collage de points.

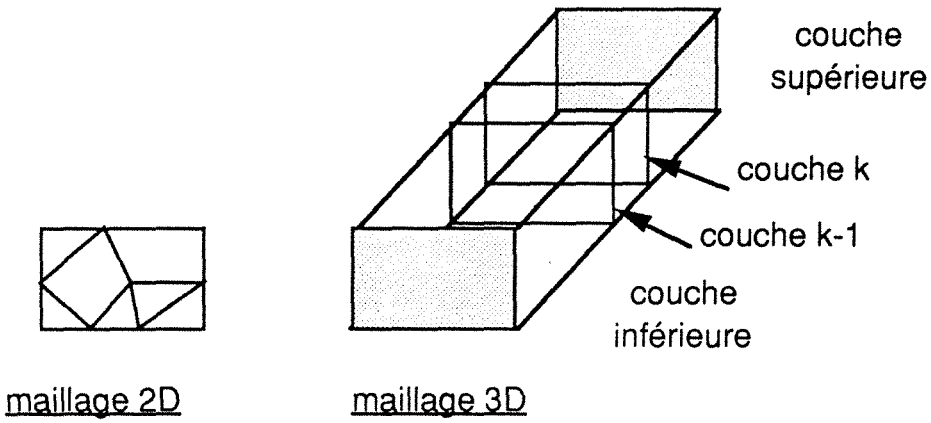


Figure IV-12 : notations pour la suite de l'exposé

IV-4-2-a : Maillage bidimensionnel

Nous partons d'un maillage bidimensionnel auquel nous imposons la contrainte suivante:

_ être composé de triangles ou de quadrilatères linéaires.
_ la forme du domaine peut être tout à fait quelconque.

Nous allons utiliser de façon intensive les données suivantes:

_ les coordonnées 2D de tous les points du maillage de base
_ la description de chaque élément par ses connections
_ L'ensemble des côtés frontières

IV-4-2-b : Transformation dans l'espace et position des couches

Nous disposons de trois types de transformations de base qui sont :

_ rotation d'axe quelconque
_ translation
_ dilatation autour d'un point donné

Ainsi que toute composée prise dans n'importe quel ordre de deux de ces transformations de base

Ces transformations ont deux buts :

1) définir la position de la première couche (couche de base) dans l'espace. Par défaut, la couche est prise dans le plan z=0, les deux autres coordonnées x et y restant les mêmes que celles du maillage 2D. Si cela ne convient pas, il est possible de placer la couche ailleurs. La transformation alors utilisée ne sert qu'a cette tâche.

2) définir le cheminement de la couche de base dans l'espace permettant la construction des couches successives et donc l'édification du maillage par empilement. A priori cette transformation n'a rien à voir avec la précédente.

Une fois fixée la position de la couche de base et le type de transformation à effectuer, il suffit à l'utilisateur d'imposer le nombre n de couches souhaitées. Pour l'instant, les n couches se trouvent répartis uniformément entre la première et la dernière couche. La possibilité de faire varier la position des couches intermédiaires est en cours de développement.

Nous proposons deux types de constructions :

- La construction globale construit la couche courante à partir de la topologie de la couche de base à tous niveaux.

- La construction locale construit la couche courante à partir de la topologie de la dernière couche crée lors de la dernière transformation.

Il est donc aussi facile de générer un maillage issu d'une seule transformation qu'un maillage issu de plusieurs. La construction locale offre un meilleur contrôle du maillage lorsqu'il y a plusieurs transformations à utiliser.

IV-4-3 : Création des éléments. Dégénérescence du maillage

Notation: Nous supposerons par la suite que nous sommes en train de construire la couche p connaissant la couche (p-1). Nous noterons i,j...l les numéros des points de l'espace de la couche p-1 et F(i),...,F(l) les numéros des points de l'espace de la couche p image des points i,...,l par la transformation F. Nous noterons α_{p-1}, ,γ_{p-1} les éléments plans de la couche p-1, α_p, ,γ_p ceux de la couche p et V_α,...,V_γ les éléments volumiques issus des éléments des deux couches.

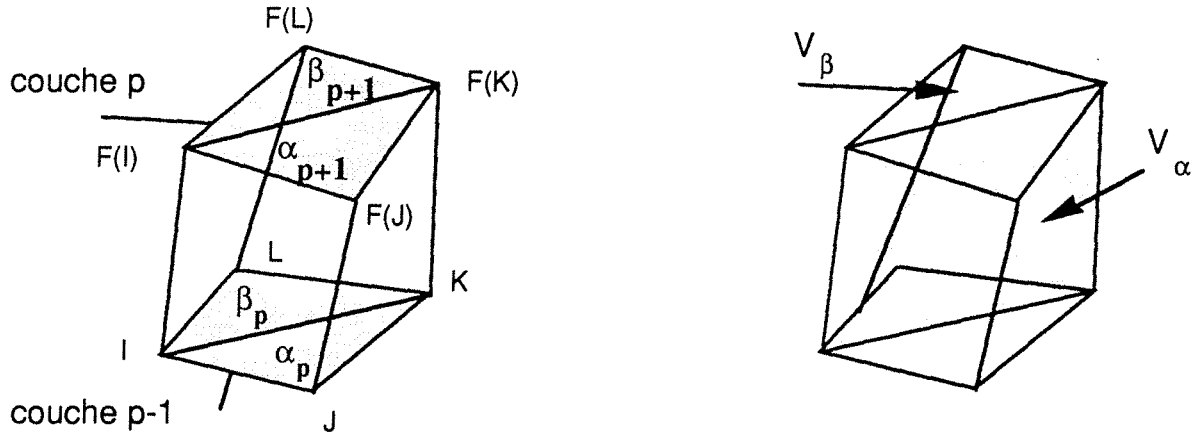


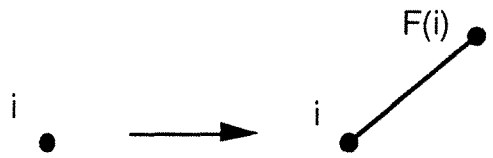
Figure IV-13 : notations de construction

Nous voyons qu'en général, l'image d'un point est un point de l'espace différent, quoiqu'il soit possible dans certains cas que ce point reste invariant. Dans le cas où un ou plusieurs points d'une entité (arête, élément) seront invariants, nous serons dans un cas dit "dégénéré" et il faudra faire attention à construire des entités ayant un sens.

Voyons maintenant l'image dans l'espace des diverses entités du plan.

Image d'un point :

_ cas normal : l'image d'un point est un segment.



_ cas dégénéré : l'image d'un point est un point.

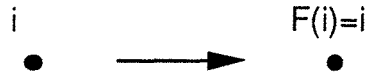
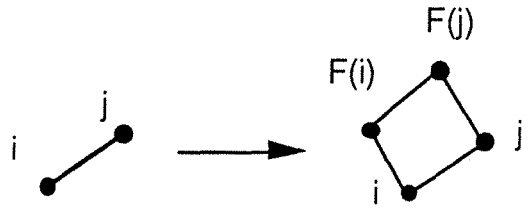
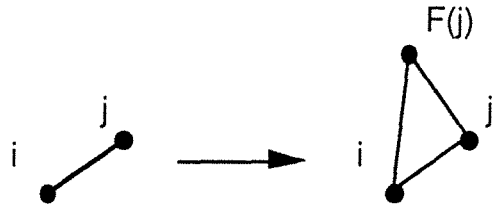


Image d'une arête :

_ cas normal : l'image d'une arête est une facette quadrangulaire.



_ cas dégénéré : un seul point dégénéré : on obtient une facette triangulaire.



_ cas dégénéré : deux points dégénérés : on obtient l'arête elle même.

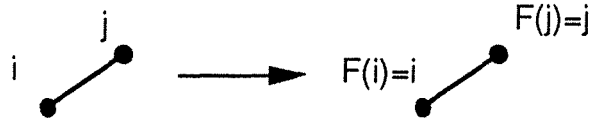
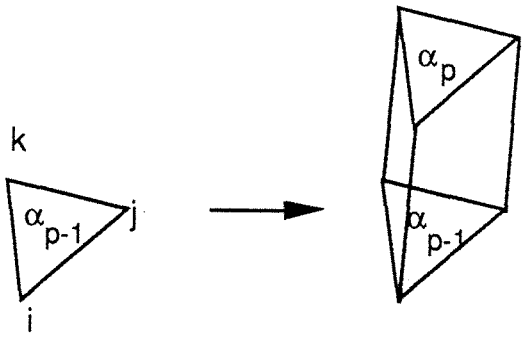


Image d'un élément triangulaire

_ cas normal : on obtient un pentaèdre



_ cas dégénéré : seul le cas suivant donne un type d'élément fini utilisable : c'est le cas de deux points dégénérés ce qui conduit à la création d'un tétraèdre. Les autres cas sont à rejeter.

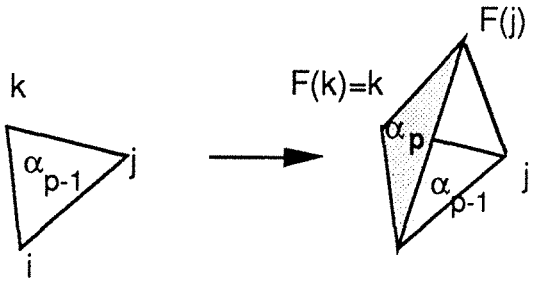
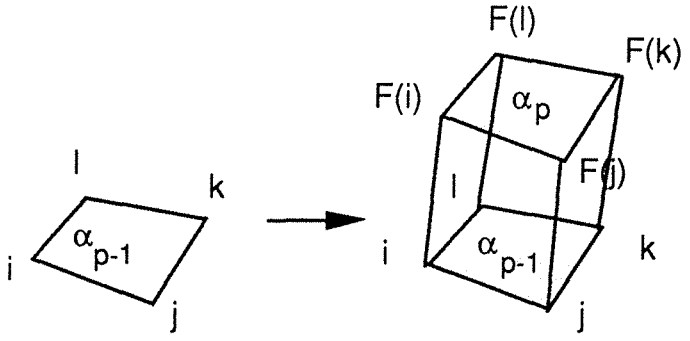
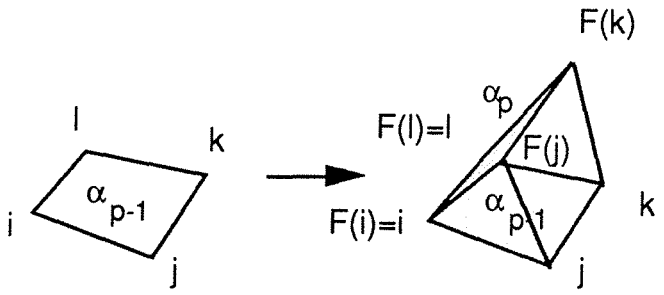


Image d'un élément quadrangulaire

_ cas normal : on obtient un hexaèdre



_ cas dégénéré : seul le cas suivant donne présente un intérêt au niveau des éléments crées ; c'est le cas de deux points consécutifs dégénérés. Cela conduit à la création d'un pentaèdre. Les autres cas sont à rejeter.



IV-4-4 : Transmission des données

A partir du maillage 2D, nous pouvons transmettre deux types d'informations concernant :

- La définition des différent types de matériaux utilisés.
- La forme du contour géométrique de la pièce.

Les numéros de référence des matériaux du maillage 3D sont déduits de la donnée des numéros de matériau du maillage 2D comme l'indique le schéma ci-dessous.

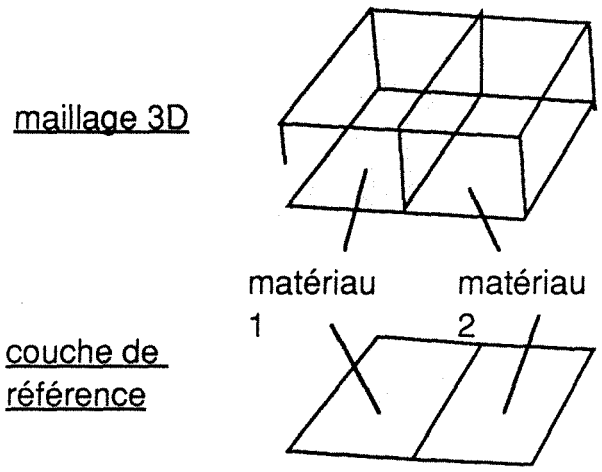


Figure IV-15 : définition des matériaux

Bien entendu, il ne s'agit là simplement que de numéros de référence. Les caractéristiques de chaque matériau ne sont définis que lors de la phase de résolution.

En ce qui concerne les frontières géométriques du domaine, nous en distinguons globalement deux :

_ les frontières supérieures et inférieures (première et dernière couche.

_ les frontières latérales du domaine.

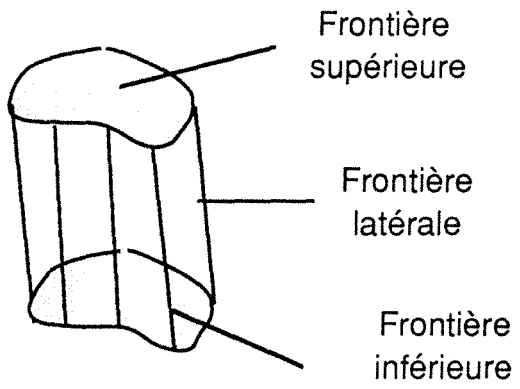


Figure IV-16 : Les frontières du domaine

Les frontières supérieures et inférieures sont généralement planes (ce n'est pas toujours le cas). Les frontières latérales reçoivent un numéro de référence correspondant à l'arête 2D qui les a engendrée.

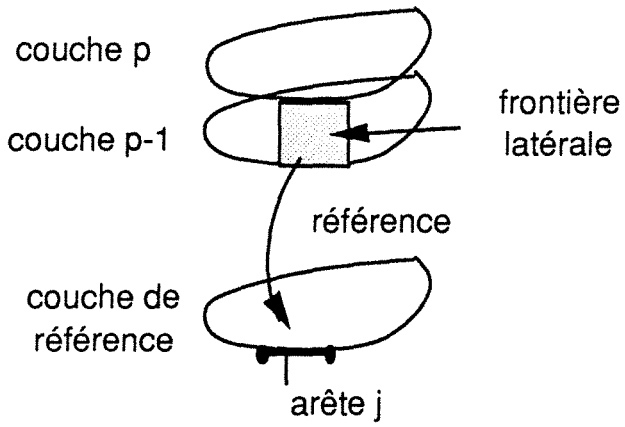


Figure IV-17 : références des facettes latérales

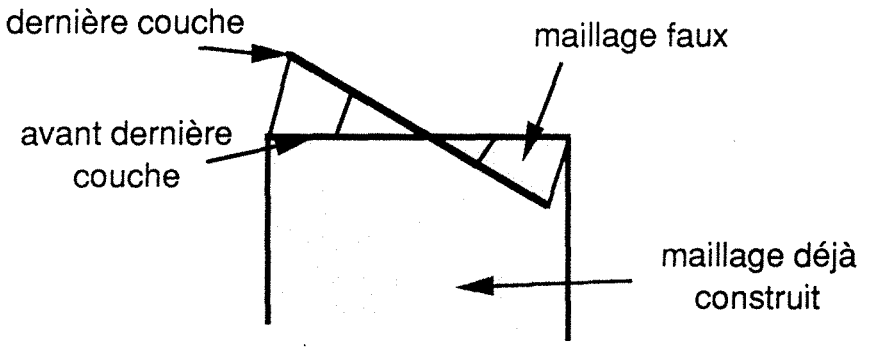
Ce type de référence permet de définir géométriquement l'identité des frontières du domaine. Si lors du maillage, une nouvelle transformation est introduite, basée sur l'une des deux couches possibles (supérieures ou inférieures) , les frontières du domaine sont automatiquement réactualisées de façon à obtenir une structure de données cohérentes.

Notre programme se limite à la référenciation des faces. Il peut être aussi nécessaire de définir les arêtes 3D. Il est possible de déduire les arêtes des facettes frontières, mais c'est un travail long et pénible.

IV-4-5 : Dégénérescence de la transformation

La définition du maillage par une telle méthode n'assure pas toujours que celui-ci soit conforme au point de vue des éléments finis. On peut citer l'exemple d'une rotation par rapport à un axe passant au milieu du maillage 2D et coplanaire avec celui-ci.

De façon générale, une construction de couche mal conçue peut provoquer des problèmes d'interpénétration de maillage difficiles et coûteux à détecter.



IV-18 : Exemple de construction fausse

[IN 6] propose une solution astucieuse pour résoudre le problème. Mais, en général, la détection d'interpénétration de maillages est délicate et coûteuse à traiter. Celle ci est généralement compensée par une gestion plus interactive du logiciel ainsi que par des aides de visualisation.

La méthode proposée [IN 6] consiste à tester la position des points crées par rapport à la couche précédente. La position de points "en dessous de la couche" indique une position critique. Le problème est résolu de la façon suivante :

_ les points "en dessus de la couche" sont construits normalement.

_ Les points "en dessous de la couche" sont considérés comme dégénérés et donc restent invariants.

_ Les points invariants sont considérés comme dégénérés, comme précédemment.

Ceci revient donc à substituer à la couche plane une nouvelle couche non plane comme le montre le schéma ci-dessous

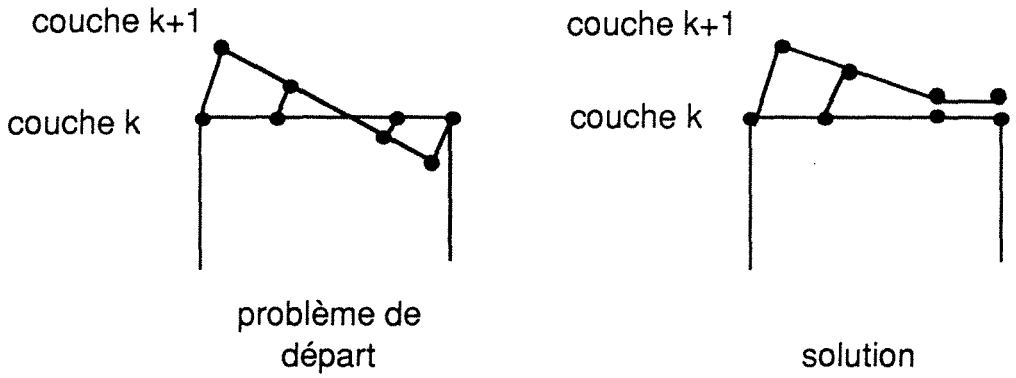


Figure IV-19 : couche dégénérée

Cette méthode a l'inconvénient de générer une couche qui n'est plus plane. Pour y remédier, on construit une "couche virtuelle" : cette "couche virtuelle" (associée à la couche k+1) se compose des points réellement calculés sans tenir compte des dégénérescences des points, ce qui peut se résumer de la façon suivante :

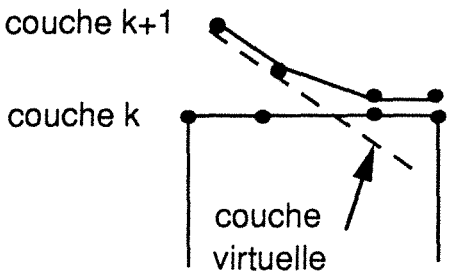


Figure IV-20 : définition de la couche virtuelle

A chaque nouvelle couche à problèmes, la couche virtuelle est réactualisée comme dans le maillage classique. La couche virtuelle est utilisée jusqu'à ce que la dégénérescence disparaisse, c'est à dire, jusqu'à ce que la couche générée soit soit plane. On revient alors au schéma classique.

En voici une illustration : à partir de la couche k, on construit une couche k+1 par rotation puis une couche k+2 par translation.

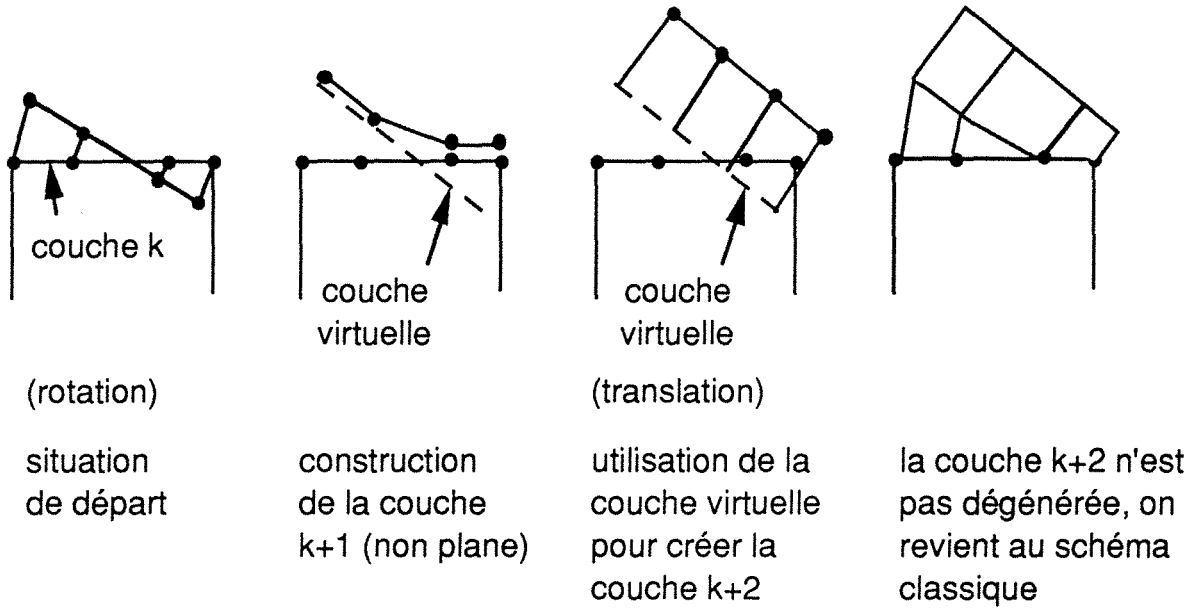


Figure IV-21 : exemple de maillage

IV-4-6 : Bouclage de maillage

Il est fréquent de trouver des maillages cycliques, c'est à dire des maillages dont les couches inférieures et supérieures sont confondues (exemple d'une rotation de 360 °). Ce cas est automatiquement détecté par le logiciel. Les faces communes sont détruites en totalité ou non selon le cas ..

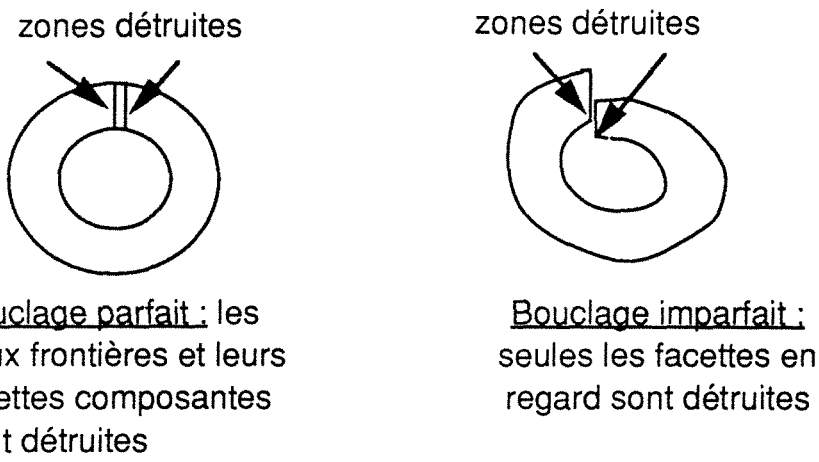


Figure IV-22 : Bouclages de maillages

Il n'est malheureusement pas possible de traiter tous les cas. En particulier, il se peut que des facettes se retrouvent coplanaires sans que tous les sommets de l'une coincident avec ceux de l'autre. Il est extrêmement difficile de traiter le problème (sauf peut être par division de facettes jusqu'à coïncidence). Le maillage construit dans ce cas est correct au sens des éléments finis et au point de vue du contour, seulement, il y a des faces internes non éliminées, ce qui ne nuît ni au maillage ni à la représentation des parties cachées.

IV-5 : calcul des parties cachées

IV-5-1 : introduction

Comme nous l'avons vu précédemment, la représentation en perspective d'une pièce tridimensionnelle peut poser quelques problèmes de clarté. Il faut d'abord éviter les déformations dues à la perspective, mais ce n'est pas un gros problème. Le plus difficile, c'est d'avoir une idée de la "profondeur de la pièce". La représentation filaire du maillage (c'est à dire le fait de ne représenter les éléments que par leurs arêtes constitutives sans idée de volume ni de matière), conduit à mettre tous les faces ou les arêtes quasiment sur le même niveau. IL est dès lors difficile d'appréhender la forme réelle de l'objet ou même de savoir "qu'est ce qui est devant, qu'est ce qui est derrière". La complexité de la pièce ou le nombre d'éléments ne font qu'aggraver le problème. En voici une illustration sur un exemple on ne peut plus trivial tiré de [GR 4] :

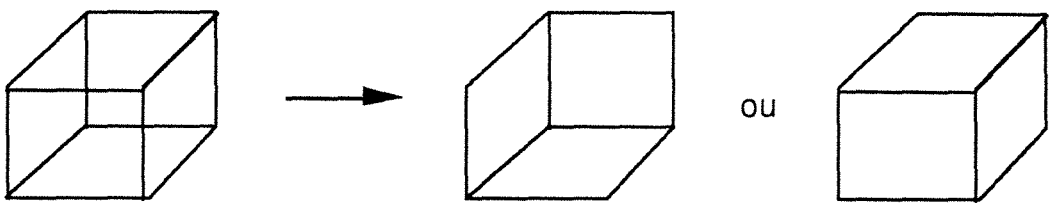


Figure IV-23

Sur l'exemple ci-dessus, on voit que la vue filaire permet deux interprétations possibles. Lorsqu'on calcule les parties cachées , l'ambiguïté disparait aussitôt.

Le calcul des parties cachées (éventuellement répété sous différents angles de vue) est donc une solution efficace au problème de la représentation plane de volumes.

IV-5-2 : Différents types de méthodes de calcul de parties cachées

Il existe de nombreuses méthodes de calcul de parties cachées. Aucune d'entre elles n'est parfaite, mais chacune possède ses avantages et ses inconvénients qui fera qu'elle sera choisie plutôt qu'une autre en vue d'une application précise.

Certains algorithmes, capables de fournir des solutions rapidement, sont utilisés pour calculer des vues en temps réel (pour des simulateurs de vol par exemple). D'autres donnent des vues réalistes et précises tenant compte, entre autres, des ombrages, des textures, des réflexions etc ..., et sont plutôt utilisés pour les images de synthèse. En général, il faut trouver un compromis entre vitesse et précision selon les impératifs que l'on s'est fixé.

La plupart des algorithmes de calcul utilisent deux notions fondamentales :

_ la notion de tri géométrique
_ la notion de cohérence de l'image

La notion de tri géométrique est utilisée pour définir un ordre de priorité parmi les objets. Le critère du tri est généralement la distance entre l'observateur et l'objet. On part du principe que plus cette distance est grande, plus l'objet aura de chance d'être occulté par d'autres objets plus rapprochés du point de vue. Ceci permet d'affecter , mais de façon imparfaite, une priorité à chaque objet, priorité d'autant plus élevée que l'objet sera proche de l'observateur. Cependant, ce simple tri n'est généralement pas suffisant pour avoir une vue correcte. L'efficacité d'une méthode de calcul dépend de l'efficacité de sa méthode de tri.

La notion de cohérence permet d'augmenter l'efficacité des méthodes employées. La cohérence traduit le fait que , localement, les caractéristiques d'un objet ou d'une image ont tendance à ne pas varier beaucoup. Ainsi, si l'on connait des propriétés sur un morceau d'entité, il est fort probable que ces propriétés sont vraies sur l'ensemble de l'entité. Cette propriété est beaucoup utilisée dans certains types d'algorithmes à balayage [GR 1] [GR 2] [GR 4].

On peut séparer les méthodes de calcul en trois selon le système de coordonnées dans lequel elles travaillent :

_ Celles qui travaillent dans le repère objet fournissent généralement un résultat parfait (à la précision de l'ordinateur). La qualité du résultat reste constant quel que soit l'agrandissement effectué. Ces méthodes sont très utilisées lorsqu'un résultat précis est exigé, comme en ingéniérie. Citons les méthodes de Roberts, Galimberti, Weiler et Atherton [GR 1] [GR 2] [GR 4].

_ Celles qui travaillent dans le repère image avec le système de coordonnées entières utilisé par l'écran. Leur principe consiste à calculer l'image pixel par pixel. De ce fait , les calculs sont effectués à la précision de l'écran ce qui permet un traitement plus rapide. En contrepartie, si l'on agrandit la vue, il est nécessaire de tout recalculer sous peine d'obtenir des résultats faux. Citons parmi ces méthodes, celles de Warnock, Watkins, Bouknight [GR 1] [GR 2] [GR 4].

_ Notons enfin l'existence de méthodes hybrides travaillant à cheval sur les deux espaces (objet et image). Elles associent des tests de priorité dans l'espace objet à des tests dans l'espace image. On peut citer les algorithmes de Schumacher et celui de Newell [GR 1] [GR 2] [GR 4]. C'est cette dernière méthode que nous utiliserons dans notre logiciel.

Les algorithmes de l'espace objet comparent chaque objet de la scène avec tous les autres. Leur complexité de calcul est donc proportionnelle à n^2 (où n est le nombre d'objet de la scène). Les algorithmes de l'espace image comparent chaque objet avec tous les pixels de l'écran. Le nombre de calcul requis est donc proportionnel à n x p (où p est le nombre de pixels de l'écran, soit en général, de l'ordre de 500^2). Les algorithmes de l'espace objet semblent donc plus rapides. Cependant, en utilisant les propriétés de cohérence de l'image on arrive à faire chuter notablement le volume des calculs. De ce fait, les calculs dans l'espace image sont généralement plus rapides.

IV-5-3 : Préliminaires au calcul des parties cachées

IV-5-3-a : Conventions

Nous désignerons par [O,x',y',z'] le repère de l'observateur. Nous allons travailler sur les facettes extérieures du domaine. Ces facettes sont automatiquement fournies par le logiciel de maillage, quelle que soit la méthode utilisée (par bloc ou par topologie cylindrique). Chaque facette est déterminée par l'ensemble de ses sommets parcourus suivant un sens déterminé. Ce sens est défini de façon à ce que la normale soit toujours orientée vers l'extérieur (Figure IV-24)

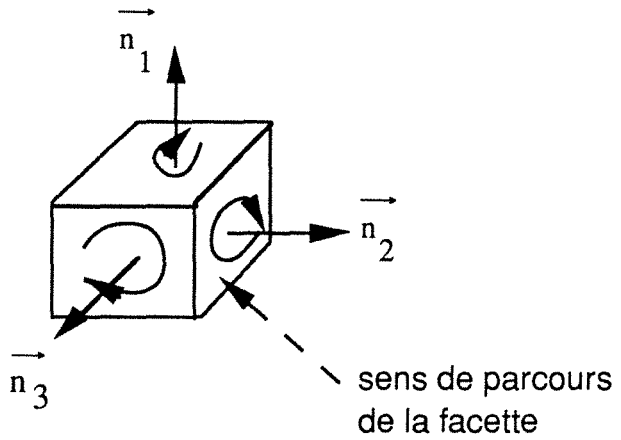


Figure IV-24 : orientation des normales

Cette notion joue un rôle important dans l'étude des parties cachées car elle donne des indications précieuses sur la visibilité de chaque facette. Cependant, pour pouvoir être utilisée, il faut que la normale à la facette soit unique, c'est à dire que la facette soit plane.

IV-5-3-b : Planéité des facettes

Dans toute la suite du programme, nous supposerons que nous n'avons affaire qu'a des facettes planes. Ceci permet donc d'être en conformité avec la remarque précédente.

De façon pratique, nos facettes sont issues d'éléments finis cubiques 8-nœuds et 20-nœuds ainsi que de pentaèdres et d'hexaèdres. Nous obtenons donc des facettes à 3, 4 ou 8 nœuds. (Figure IV-25)

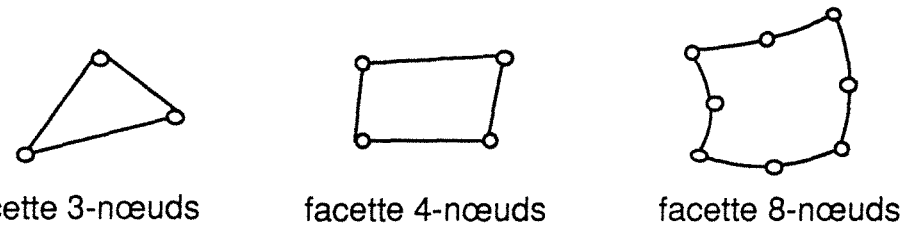


Figure IV-25

La facette 3-nœuds est plane, ce qui résoud le problème. Pour les deux autres, nous subdivisons en sous facettes planes à 3-nœuds de la façon suivante :

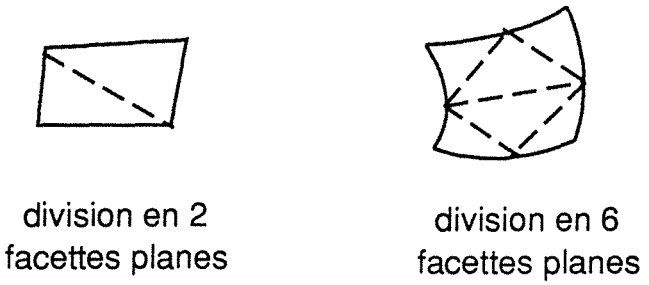


Figure IV-26 : décomposition en sous facettes planes

Ce découpage a l'avantage de ne nécessiter le calcul d'aucun point supplémentaire. Si d'aventure, la facette est déjà plane, celle est conservée sous sa forme originale afin de diminuer le nombre des calculs.

IV-5-4 : Algorithme du peintre. Premiers calculs

IV-5-4-a : Principe de la méthode

L'idée de cet algorithme s'apparente à la façon dont travaille un peintre (d'où son nom). En peignant de l'arrière plan jusqu'au premier plan, chaque couche de peinture vient recouvrir les précédentes. Ainsi, seules demeurent visibles la dernière couche ou toute couche non recouverte.

Nous allons procéder de façon analogue avec nos facettes. Nous supposons que nous pouvons leur affecter un numéro de priorité, les faces à afficher à l'arrière plan ayant les numéros de priorité les moins élevés. Nous affichons les facettes les unes après les autres dans l'ordre de leurs priorité de façon que chaque facette recouvre sa surface de façon opaque. Nous obtenons le résultat suivant :

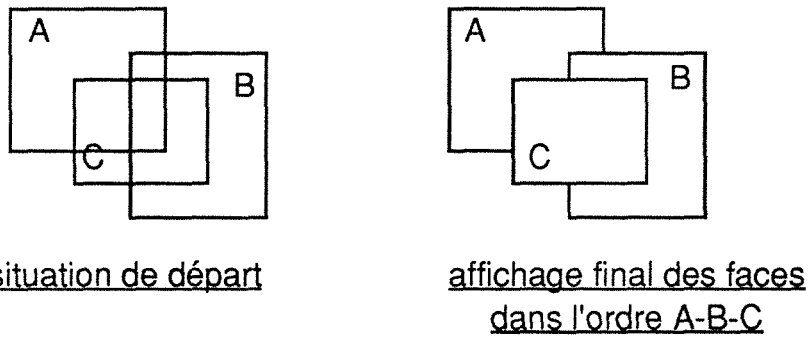


Figure IV-27 : principe de la méthode du peintre

IV-5-4-b : algorithme général

La difficulté principale pour établir notre algorithme consiste à déterminer l'ordre de priorité de chaque facette de telle manière que l'affichage suivant cet ordre donne un résultat correct. Nous avons choisi pour cela la méthode de N Newell, R Newell et T Sancha de l'université de Cambridge. Cette méthode se décompose en quatre étapes.

1) . Mise en place des données.
2) . Elimination des faces arrières.
3) . Tri des faces suivant leur côte z maximale établissant ainsi un ordre de priorité initial destiné à être modifié.
4) . Levée des ambiguïtés restantes pour obtenir les priorités définitives des facettes.
5) . Affichage puis remplissage de chaque facettes selon l'ordre des priorités.

Revenons maintenant en détail sur ces phases :

L'étape 1 se réduit à la vérification de la planéité de chaque facette d'élément fini. Si ce n'est pas le cas, il y a subdivision selon le schéma vu précédemment.

L'étape 2 sert à simplifier le travail en diminuant le nombre des facettes sur lesquelles on doit travailler. Cette simplification est basée sur le fait suivant : toute facette séparée de l'observateur par un morceau ou la totalité de la pièce n'est pas visible (figure IV-28), cela permet donc de les éliminer du nombre des facettes :

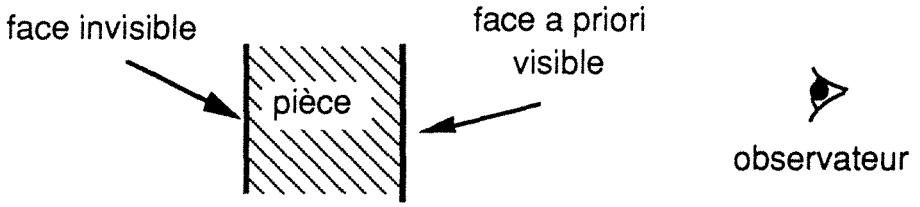


Figure IV-28 : facettes visibles et invisibles

D'un point de vue pratique, nous définissons pour chaque facette un vecteur de vision reliant l'observateur avec n'importe lequel des sommets de la facette. Nous faisons alors le produit scalaire entre ce vecteur de vision et le vecteur normal sortant de la facette.

Deux cas se présentent alors :

_ Le produit scalaire est positif ou nul; la facette est invisible, on n'en tient plus compte.

_ Le produit scalaire est négatif; la facette est a priori visible, on la conserve.

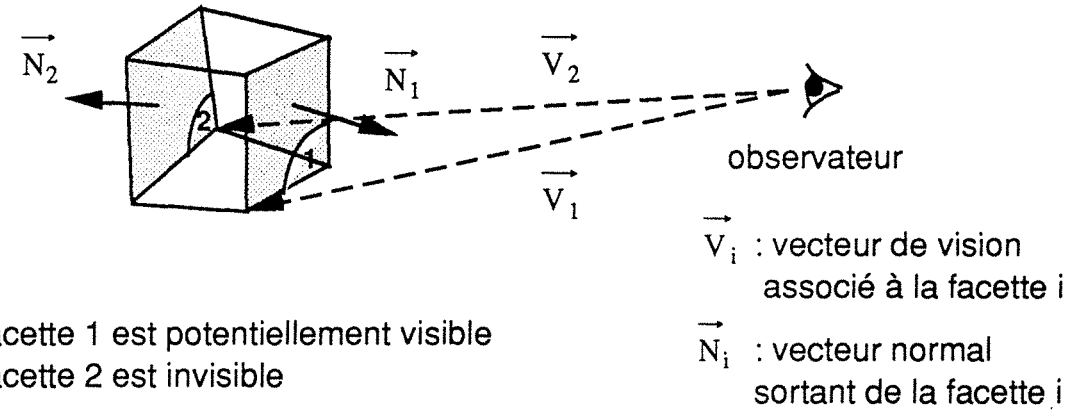


Figure IV-29 : visibilité de facettes

Ceci appelle deux remarques :

_ Le fait qu'une facette soit potentiellement visible n'implique pas que celle-ci soit effectivement visible, même partiellement (c'est le cas de pièces non convexes).

_ Une remarque liée à la précédente : si la pièce est convexe, ce simple test suffit à éliminer toutes les parties cachées. On peut alors afficher les facettes restantes dans n'importe autre ordre, le résultat sera correct.

L'étape 3 pose un problème classique de tri suivant la côte z maximale (ie le maximum des z pris sur tous les sommets de la facette) sur toutes les facettes restantes. Le nombre de ces dernières pouvant être encore élevé, nous avons choisi un tri bulle. Le tri permet d'obtenir un ordre de priorité donnant déjà une idée de l'ordre réel. Il n'est cependant généralement pas correct et est destiné à être modifié par l'étape 4. Cette étape, véritable cheville ouvrière de la méthode va être développée en détail dans le paragraphe suivant.

Enfin, l'étape 5 est un simple problème d'affichage et de remplissage de polygones à l'écran. Il ne nous semble pas nécessaire de nous appesantir.

IV-5-5 : Algorithme central de levée des ambiguïtés

Il s'agit là d'un algorithme itératif de comparaisons des facettes entre elles visant à établir un ordre de priorité correct. Il se compose d'une série de test géométriques simples.

Par la suite, nous supposerons que nous avons à comparer la facette de priorité i à toutes les facettes de priorité supérieure. Ou bien le résultat est correct, c'est à dire que la facette de priorité i est bien "derrière" celles de priorités supérieures, et donc les priorités restent inchangées, ou bien le test est incorrect, et dans ce cas, il y a permutation de priorités. En ce sens, le processus s'apparente un peu au tri bulle.

Voyons maintenant en détail ces tests :

Nous comparons la facette de priorité i à celle de priorité j (j>i).

_ test 1 : disjonction suivant les z' ?

_ test 2 : disjonction suivant les x' ?

_ test 3 : disjonction suivant les y ?

_ test 4 : tous les points de la facette j sont-ils du même côté du plan défini par la facette i que l'observateur ?.'

_ test 5 : tous les points de la facette i sont-ils du côté du plan défini par la facette j opposé à l'observateur ?.'

_ test 6 et 7 : on permute les priorité i et j et on réitère les tests 4 et 5.

Dans les cas où tous ces tests ont échoué, on procède à la coupure d'une des faces en deux pour trouver une solution au problème. Revenons maintenant en détail sur chacun de ces 7 tests.

Test 1

Nous calculons pour les facettes de priorité i et j les valeurs maximales et minimales de la coordonnée z' que nous notons $z\text{-}max_i$, $z\text{-}min_i$, $zmax_j$ et $zmin_j$. Il est évident que si $z\text{-}max_i > z\text{-}min_j$ alors, les deux facettes sont classées dans le bon ordre (figure IV-30). Si ce n'est pas le cas, le test échoue.

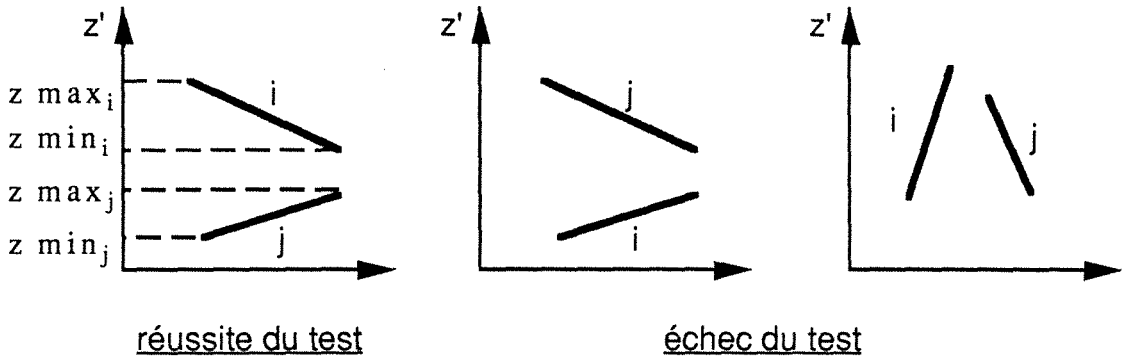


Figure IV-30 : test 1

Test 2 et 3

Ces deux tests ont la même philosophie. Ils sont effectués dans l'espace image caractérisé par ses coordonnées x_i et y_i. Nous allons regarder si les deux facettes sont disjointes selon les x_i ou les y_i. Si c'est le cas, cela signifie que l'on peut afficher la facette de priorité i indépendamment de celle de priorité j, donc que l'ordre est correct (figure IV-31). Si l'un des tests est réussi, on ne touche pas aux priorités.

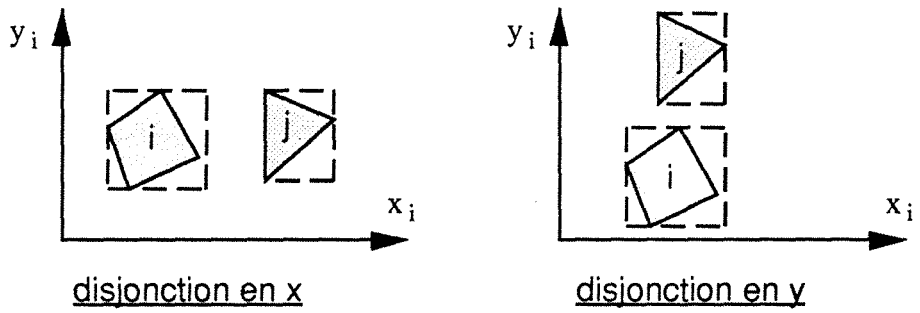


Figure IV-31 : test 2 et 3

Test 4

Les trois premiers tests ayant échoué, nous allons maintenant essayer de voir si tous les sommets de la facette j sont du même côté du plan défini par la facette i que l'observateur, soit le cas suivant (figure IV-32).

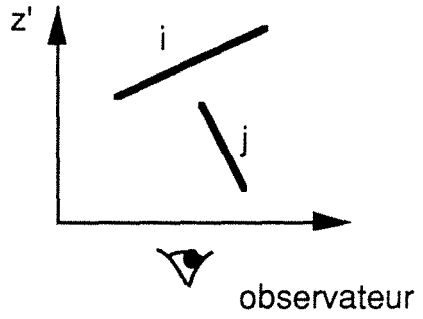


Figure IV-32 : test 4

On peut dans ce cas conclure que les priorité des deux facettes sont correctes. D'un point de vue géométrique, il suffit de tester le signe de chaque sommet de la facette de priorité j par rapport à l'équation du plan défini par la facette de priorité i. Si le test est positif, les priorités des deux facettes sont correctes.

Test 5

Ce test correspond au cas suivant (figure IV-33).

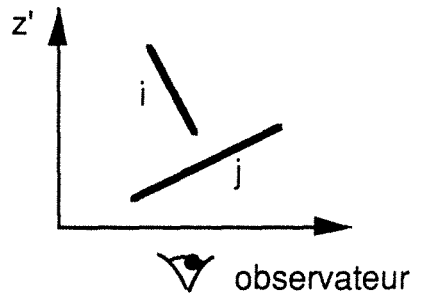


Figure IV-33 : test 5

Il faut donc vérifier que tous les points de la facette i sont du côté du plan défini par la facette j opposé à l'observateur Le traitement géométrique est analogue à celui du test précédent.

Maintenant, si les tests 1 à 5 ont échoué, il nous faut conclure que l'hypothèse selon laquelle la facette de priorité i doit être classé avant la facette de priorité j (c'est à dire en fait i>j) est à rejeter. Nous échangeons donc les priorité des faces et nous reprenons les tests 3 et 4. Nous pouvons en effet être dans les deux cas suivants :

Cas associé au test 6

Nous allons tester (test N° 6) si tous les sommets de la facette i sont du même côté du plan défini par la facette j que l'observateur. Ceci est exactement la formulation du test 4 en inversant les facettes de priorités i et j ; soit le cas suivant (figure IV-34) :

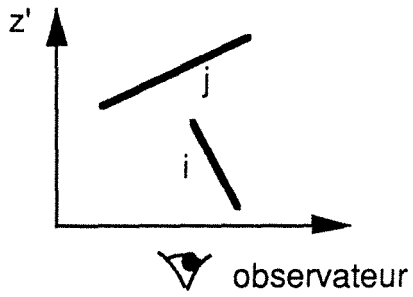


Figure IV-34 : test 6

Cas associé au test 7

Nous allons tester (test N° 7) si tous les sommets de la facette j sont du côté du plan défini par la facette i opposé à l'observateur. Comme précédemment, nous constatons qu'il s'agit d'une formulation identique à celle du test 5 en inversant les facettes de priorités i et j ; soit le cas suivant (figure IV-35) :

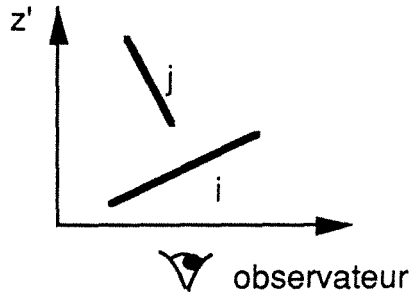


Figure IV-35 : test 7

En résumé donc, si les tests 1 à 5 ont échoué, nous procédons aux tests 6 et 7. Si ces tests réussissent, nous pouvons inverser les priorités des deux facettes et nous reprenons au début l'ensemble des tests destinés à classer la nouvelle facette de priorité i.

Il convient cependant de faire attention à ce que l'on fait, la relation d'ordre servant à trier les facettes n'est pas une relation d'ordre totale. Il se peut qu'aucun des 7 tests ne soit positifs par exemple lors recouvrement cycliques de facettes entre-elles (Figure IV-36).

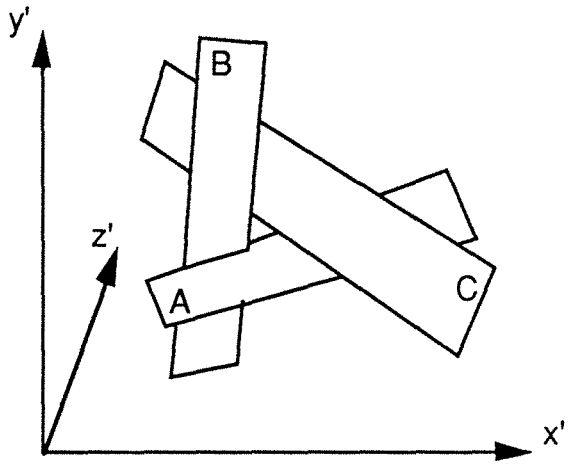


Figure IV-36 : recouvrement cyclique

Il y a donc dans ce cas bouclage du programme. Le problème se présente sous deux aspects :

_ a) détecter des éventuels bouclages du programme.

_ b) trouver une solution géométrique au problème

Le premier aspect est résolu de façon simple : lorsque nous devons permuter deux facettes un pointeur initialement à 0 est mis à 1 pour chacune des deux facettes. Ceci permet ultérieurement de repérer la re-permutation de deux facettes déjà déplacées et donc de l'interdire. Dès qu'une facette a été classée, ces pointeurs sont tous remis à 0.

Le second aspect est résolu par division de facette. Le problème n'ayant pas de solution tel quel, nous allons procéder à la coupure de la facette i en deux suivant la trace du plan défini par la facette j (figure IV-37).

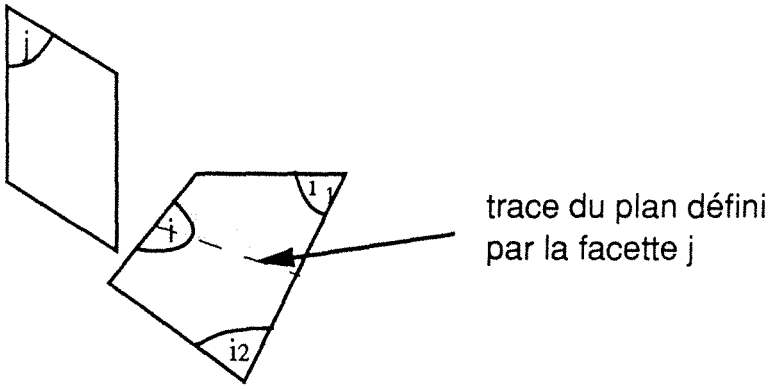


Figure IV-37 : division de la facette i

Cela définit donc deux nouvelles facettes que nous appelerons i_1 et i_2 (planes, bien entendu). De toute évidence,par construction, la facette i_1 a une priorité inférieure à la facette j et la facette i_2 a une priorité supérieure. Nous laissons donc i_1 à la place précédemment occupée par i. La facette i_2 est crée et stockée en fin de la liste de toutes les facettes. Il y a donc création de nouveaux points et d'une nouvelle facette ; D'un point de vue général, ce processus de divisions peut se répéter plusieurs fois durant le programme, ce qui engendre la création de points et de facettes artificiels (par rapport au maillage). L'ensemble de ces entités est stocké à part de façon à ne pas interférer avec la maillage.

L'ensemble de ces 8 cas permet de traiter l'ensemble des situations géométriques possibles et donc d'établir un ordre de priorité correct. Il suffit dès lors d'afficher dans l'ordre des priorités chaque facette et d'opacifier son contenu pour obtenir le résultat final.

Le résultat peut être zoomé autant de fois que souhaité (à la précision du graphisme de l'écran) sans nécessiter aucun nouveau calcul, c'est là l'intérêt du modèle quasi-exact.Il peut cependant apparaître sur la figure quelques petits segments issus des éventuelles divisions de facettes. Comme il l'a été indiqué plus haut, ces segments ne sont qu'un artifice de représentation.

D'un point de vue géométrique, nous avons la plupart du temps affaire à des facettes 3 ou 4 nœuds planes. Les critères de conformité de maillage par éléments finis permettent de limiter au maximum les "difformités" des facettes. Ceci permet de limiter les calculs, notamment lors des phases d'intersection de facettes.

IV-5-6 : Une variante

Les calculs de comparaisons de facettes sont faits sommet par sommet , ce qui représente un grand nombre de calculs et pose parfois quelques problèmes numériques (points communs à deux facettes par exemple). Une variante intéressante propose d'effectuer les tests 4 à 7 en se limitant au centre de gravité de chaque facette. Cette variante peut se justifier par la forme simple des entités mises en jeu (facettes planes et convexes). Le volume de calculs _ et donc le temps de calcul _ diminue fortement. Un tel processus n'est pas sûr à 100 % mais il donne la bonne solution dans un grand nombre de cas ; Nous l'avons donc inclus en option du logiciel.

On peut résumer notre programme par l'algorithme suivant :

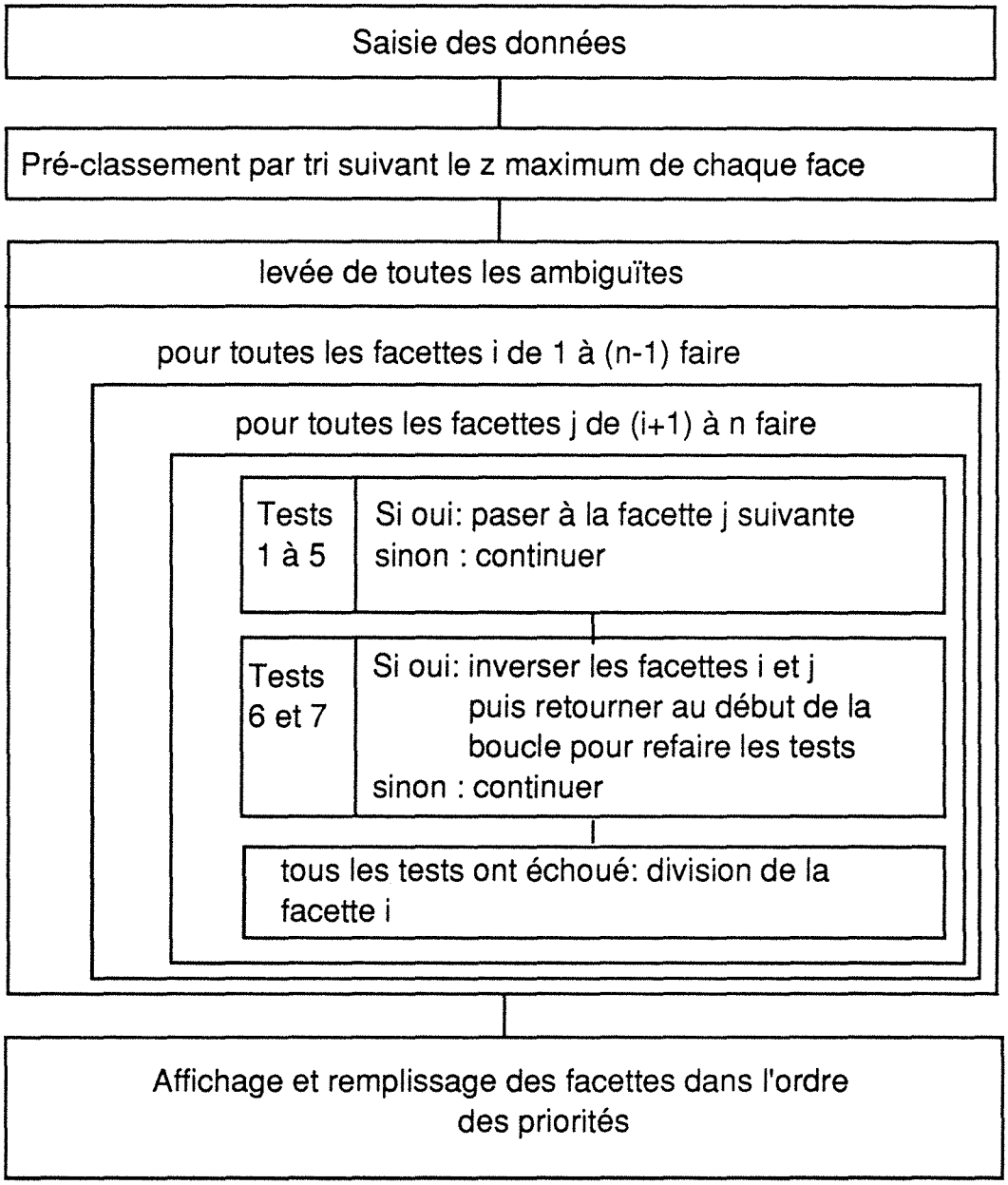


Figure IV-38 : déroulement du calcul des parties cachées

IV-6 : Algorithme général de maillage

Il nous est maintenant possible de présenter l'algorithme général de notre logiciel de maillage

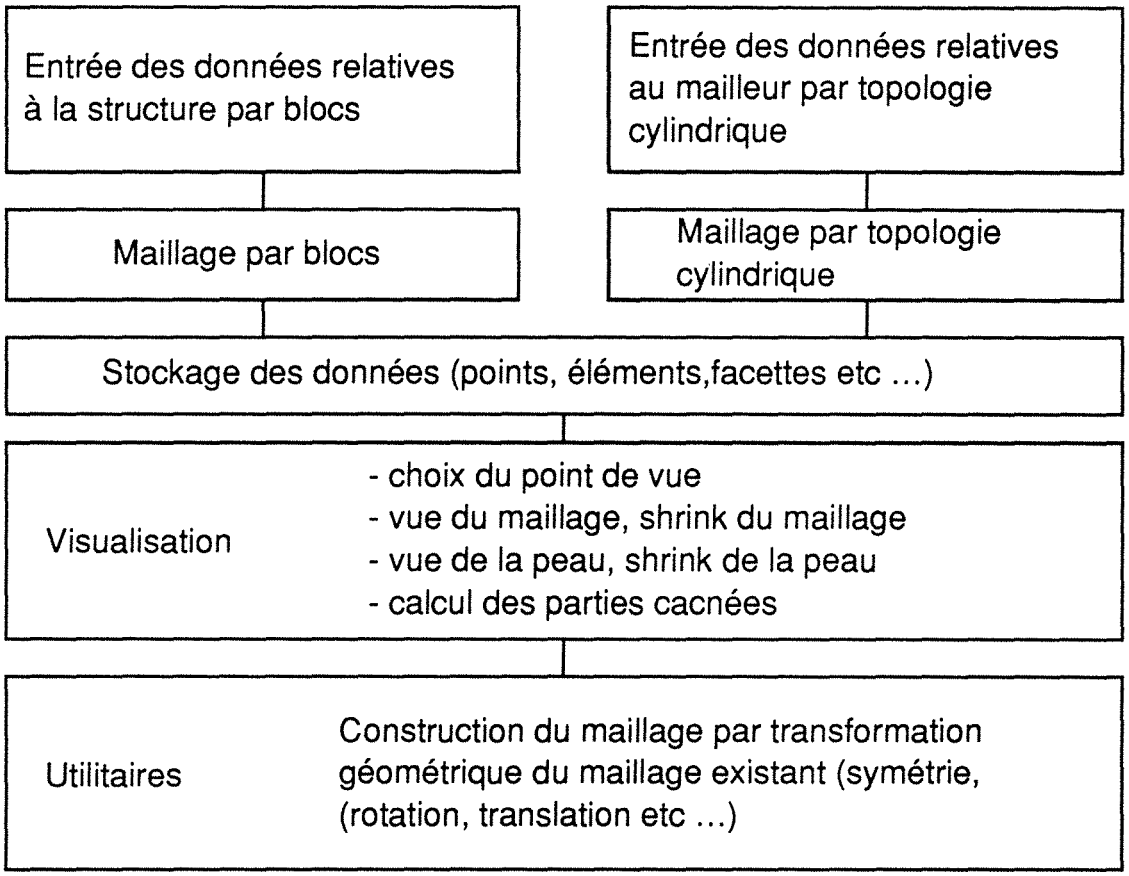


Figure IV-39 : Algorithme général du logiciel

L'ensemble des informations, tant en entrée qu'en sortie, est stockée exactement comme nous l'avons indique en début de ce chapitre. Les modules de visualisation utilisent tous la représentation en perspective selon un point de vue fixé par l'utilisateur. Il est possible de représenter le maillage ou la peau du domaine en vue filaire ou en mode shrink (contraction d'une entité par rapport à son centre de gravité).

Pour faciliter au maximum l'entrée des données, notamment pour le mailleur par blocs, nous utilisons des saisies directes à l'écran gérées par des menus déroulants. Nous estimons que ceci aide beaucoup l'utilisateur même si cela peut parfois poser quelques problèmes (non unicité de l'image écran d'un point quelconque). Lors de l'entrée des données, le nombre d'entités (points, blocs ...) reste suffisamment faible pour que l'utilisateur puisse distinguer quelque chose.

Par contre, une fois le maillage effectué, ce genre de saisie est quasiment impossible car le grand nombre d'entités intervenant rend leur distinction très délicate. Nous avons donc limité nos utilitaires de post-traitement à la construction de nouvelles parties de maillage par transformations géométriques de parties existant déjà.

Comme dans les autres programmes, nous ne pouvons dupliquer que l'ensemble du maillage par l'intermédiaire d'un motif stockant le maillage et son contour. Par la suite, il est possible, soit de conserver le motif de base, soit de la réactualiser sur le nouveau maillage construit.

Bien entendu, toute construction engendre automatiquement la création et la destruction d'entités diverses (faces et facettes frontières) de façon à rendre le résultat toujours compatible avec notre structure de données.

D'autres utilitaires sont bien sûr définissables, mais leur contrôle est beaucoup plus délicat, surtout si l'on souhaite un processus totalement automatique. Il faudrait dans ce cas utiliser des représentations de données plus variées telles que les projections sur des plans fixés, des coupes du domaine ..., bien qu'aucune d'entre elles ne soit parfaite.

IV-7 : Perspectives de développement des mailleurs 3D

Bien qu'offrant déjà un éventail de possibilités assez vaste, il est évident que notre mailleur 3D est loin de couvrir tous les problèmes possibles. De nombreux utilitaires sont à rajouter pour accroître, soit les performances de maillage, soit l'accessibilité du logiciel. Ces directions de travail offrent beaucoup de possibilités. Il est possible dans le mailleur par bloc d'étendre la notion de bloc à des formes plus variées en y adjoignant les tétraèdres et les pentaèdres à bords courbes (tétraèdres 10-nœuds et pentaèdres 15-nœuds). L'extension du catalogue de formes permet une décomposition des domaines à mailler beaucoup plus aisée. Cependant, on obtient alors un maillage pouvant être constitué d'éléments de type différents, ce qui nécessite un solveur adapté à ce genre de structure ou bien de mailler le domaine avec le type d'élément le plus pauvre (généralement le tétraèdre).

Le maillage par topologie cylindrique peut, lui aussi présenter des axes de développement intéressant, notamment par l'inclusion des éléments quadratiques dans la couche de départ. Il sera aussi possible de contrôler encore mieux la création des diverses couches évitant ainsi les problèmes de dégénérescence. Il nous apparait cependant que ce logiciel atteindra rapidement ses limites et que tout ajout de processus de contrôle ne fera que pénaliser les performances du programme.

De façon plus générale, ces algorithmes ont leur limites dans le sens où ils ne seront jamais totalement automatiques. L'élaboration d'un logiciel automatique (ou du moins , le plus possible) passe par la maîtrise de trois points essentiels :

1) L'entrée et le contrôle des données et particulièrement de la forme du domaine. Ce point doit être traité de la manière la plus interactive possible.

2) L'algorithme de maillage doit être performant et souple d'emploi. Il doit aussi permettre de traiter automatiquement le domaine fourni.

3) La représentation des résultats. Il s'agit là d'un point important qui permet de contrôler ou de saisir visuellement toutes les entités composant le maillage ou le domaine. Les projections en perspective, les représentations en mode solide, les coupes suivant des plans constituent les outils les plus utilisés.

En ce qui concerne le premier point, il semble que certaines méthodes de CAO soient prométteuses. En particulier, les générations automatiques de nappes 3D par les méthodes de Bézier ou par les fonctions splines à partir de points de contrôles peuvent permettre de construire relativement rapidement le contour d'un domaine. Ainsi définie, la nappe morceau du contour offre à la fois une approche géométrique continue du contour mais aussi la possibilité de générer selon le souhait de l'utilisateur toute une série de points frontières sur lesquels le maillage pourra s'appuyer.

Pour pouvoir répondre à cette structure de données très générale, il faut que l'algorithme de maillage soit le plus universel possible. Il semble actuellement à cet égard que les méthode les plus efficaces soient celles de Delaunay/Voronoï. Celles-ci présentent l'inconvénient de ne mailler qu'en tétraèdres. On rencontre de plus les problèmes de respect du contour tout à fait analogues à ceux rencontrés en 2D, mais plus délicats à traiter .(notons cependant que la technique de forçage peut s'étendre aux faces et a été mis en œuvre avec succès par les équipes de l'INRIA [IN 2]).

Le troisième point est certainement le plus délicat à ajuster. Soulignons d'abord, et une fois de plus son importance pour l'accessibilité du logiciel. A notre avis, les problèmes de maillage par éléments finis nécessitent des calculs aussi précis que possibles, ce qui semble exclure les processus de représentation dans l'espace image. L'inconvénient est que si les processus dans l'espace objet sont précis, ils sont cependant coûteux en temps de calcul, notamment sur micro-ordinateur. Il est certes possible d'améliorer la rapidité des calculs en affinant les algorithmes (par exemple, l'algorithme de Schumacher basé sur des principes assez semblables à ceux de l'algorithme Newell-Newell-Sancha permet une accélération des calculs). On reste cependant limité par le volume des calculs ainsi que par le manque de souplesse informatique du langage utilisé (le FORTRAN). Les progrès fulgurants de la micro-informatique devraient cependant améliorer cet état de fait.

IV-8 : Exemples

IV-8-1 : Exemple des trois quarts d'une tête de piston

A partir d'un maillage de Delaunay, nous avons maillé cette tête de piston par rotation autour d'un axe. Le maillage de la couche de base est réalisé avec des quadrilatères 4-nœuds. On notera que la pièce ne devient vraiment compréhensible qu'avec le calcul des parties cachées (figure 40-c), la vue filaire étant quasiment inutilisable (Figure 40-a).

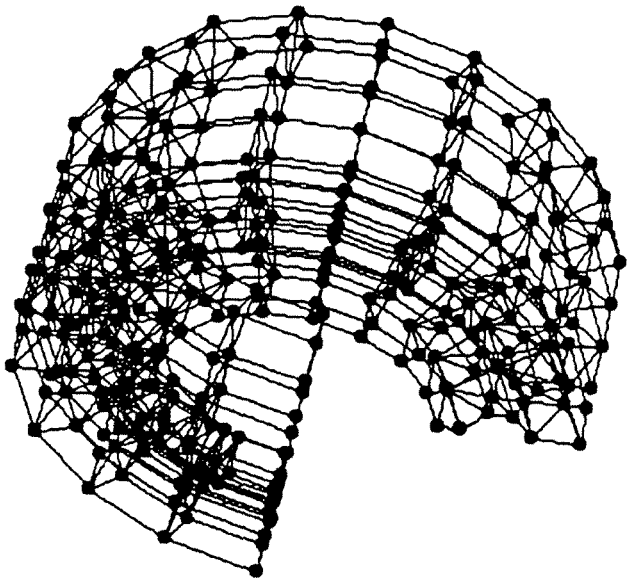


Figure IV-40-a : la vue filaire

322 nœuds, 312 éléments

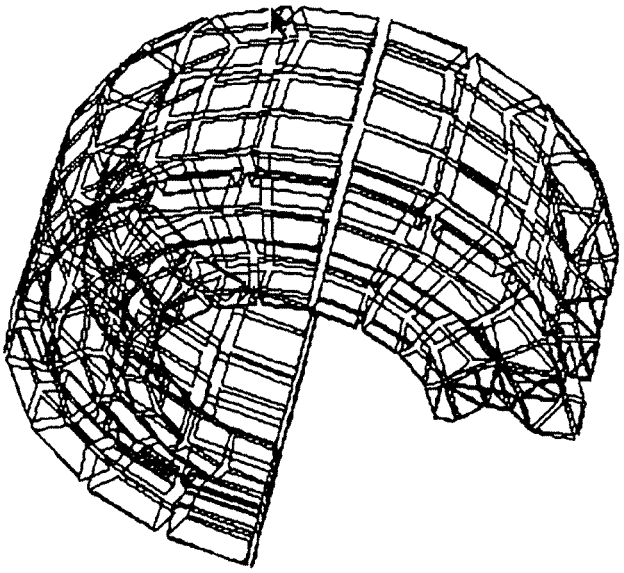


Figure IV-40-b : la peau du domaine (shrinkée)

C'est déjà un peu plus clair

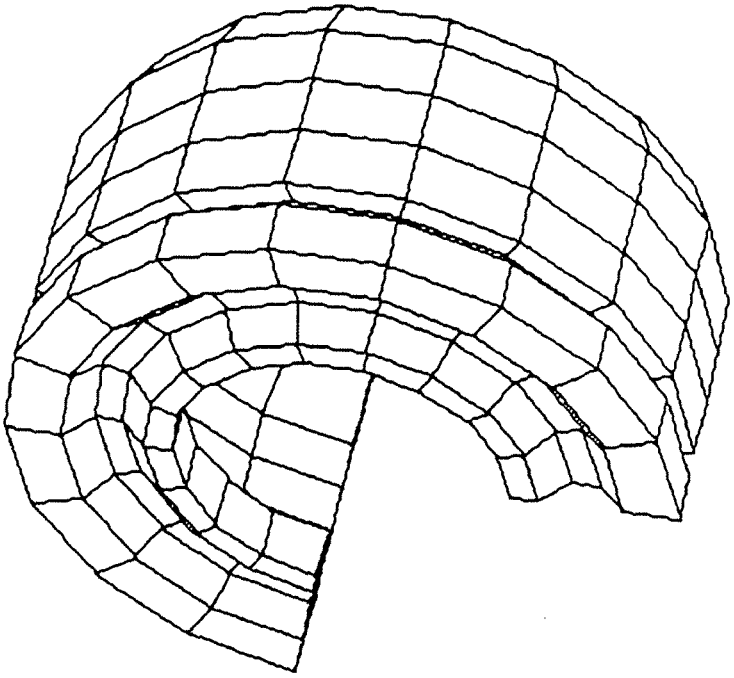


Figure IV-40-c : sans les parties cachées
On peut visualiser la pièce (environ 30 minutes sur MAC II ,tout compris)

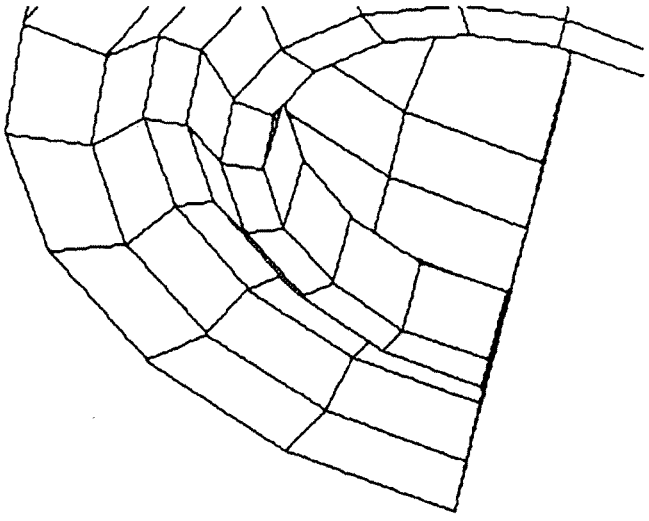


Figure IV-40-d : Un zoom du coin bas gauche

IV-8-2 : "Un quart de melon"

Maillé à partir d'un simple couche en forme de quart d'anneau que l'on fait tourner autour de l'un de ses axes. Le maillage de la couche de base est réalisé avec des quadrilatères 4-nœuds. On notera la construction d'éléments dégénérés (des pentaèdres) au voisinage de l'axe.

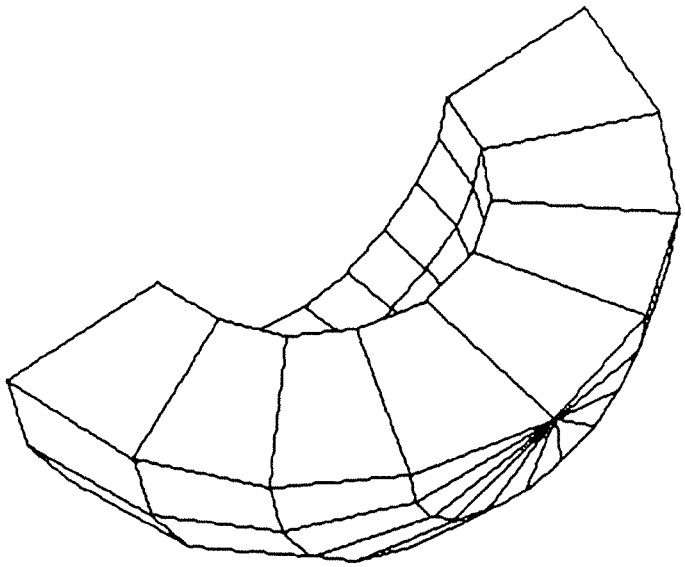


Figure IV-41 : Mailage du domaine.

IV-8-3 : Un engranage hélicoïdal

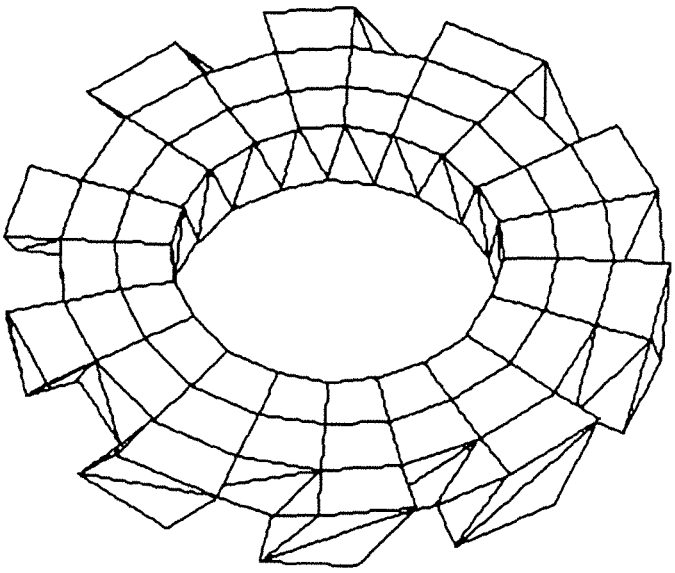


Figure IV-42

IV-8-4 : Torsion d'une poutre avec encoche

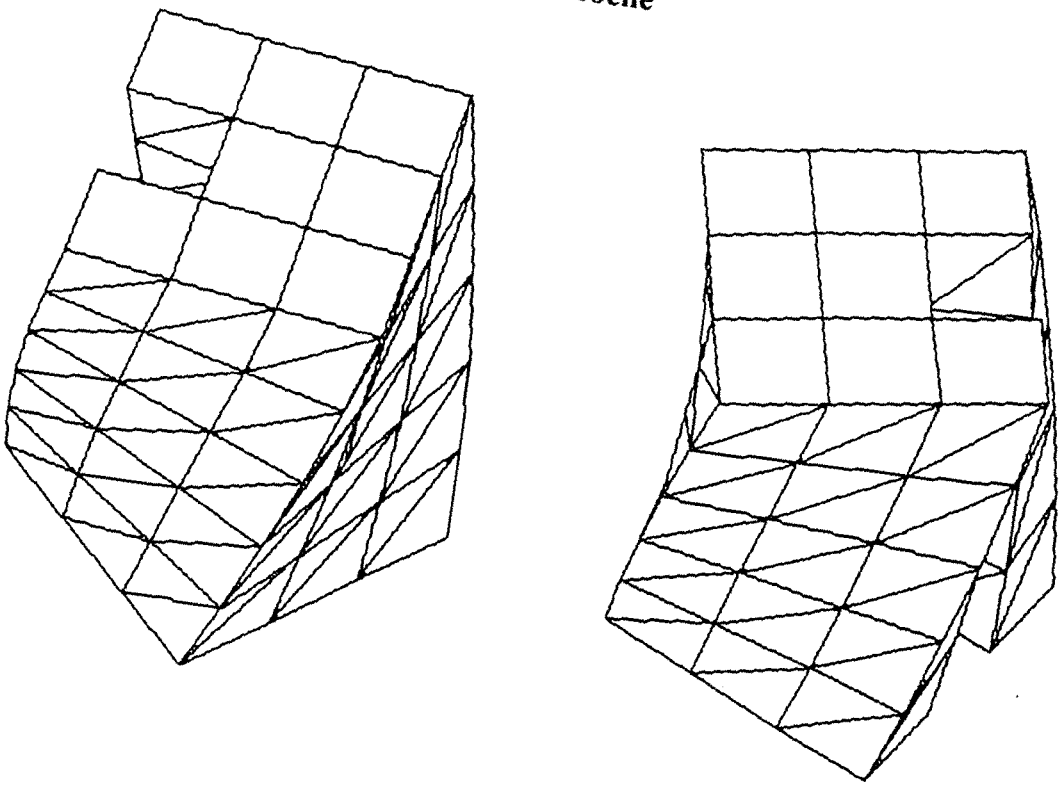


Figure IV-43

IV-8-5 : Un exemple pour le plaisir : 2 tores

cet exemple n'a aucune réalité au sens des éléments finis, il sert plutôt à illustrer les possibilités de visualisation d'un solide 3D dans l'espace.

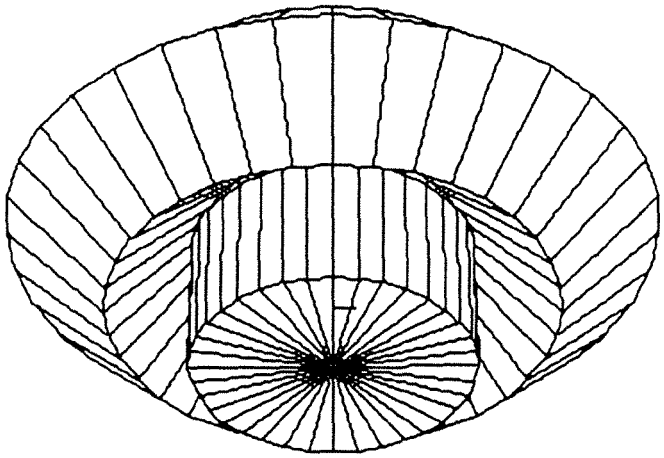


Figure IV-44-a

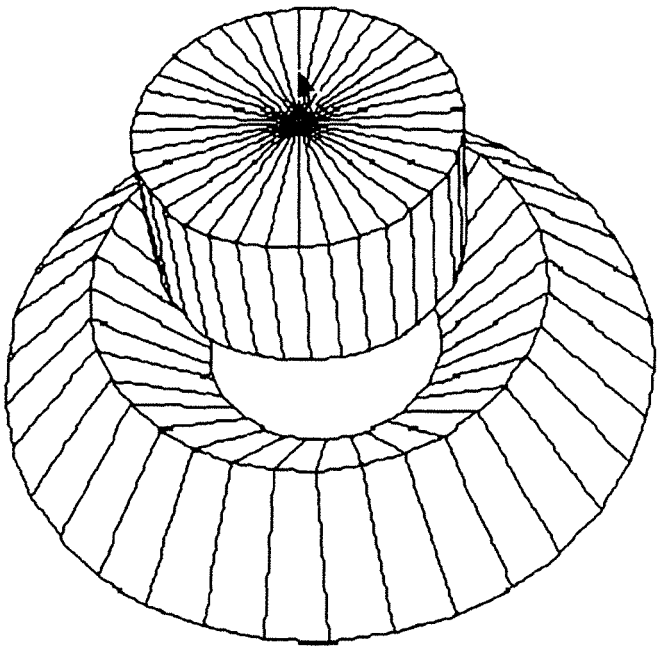


Figure IV-44-b

IV-8-6 : Maillage d'une demi-bielle

Nous utilisons le mailleur par blocs pour mailler une demi-bielle. Le domaine est décomposé en 7 blocs et 92 nœuds (figure IV-45a). La définition de ce contour relativement complexe a été facilité par des utilitaires de CAO (symétrie, recentrage de l'écran, saisie directe à l'écran, zoom). Il faut cependant noter que même avec toutes ces aides, la définition du contour a nécessité environ 20 minutes sur Mac II.

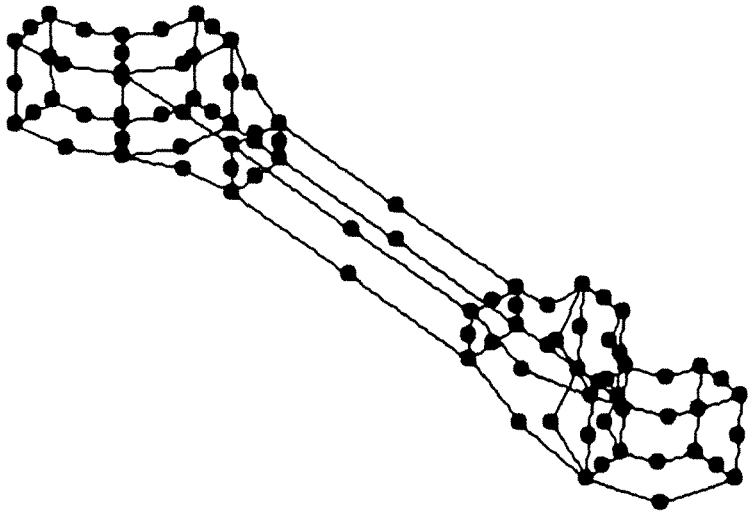


Figure IV-45-a : Définition de domaine à mailler
(7 blocs 20-nœuds et 92 nœuds)

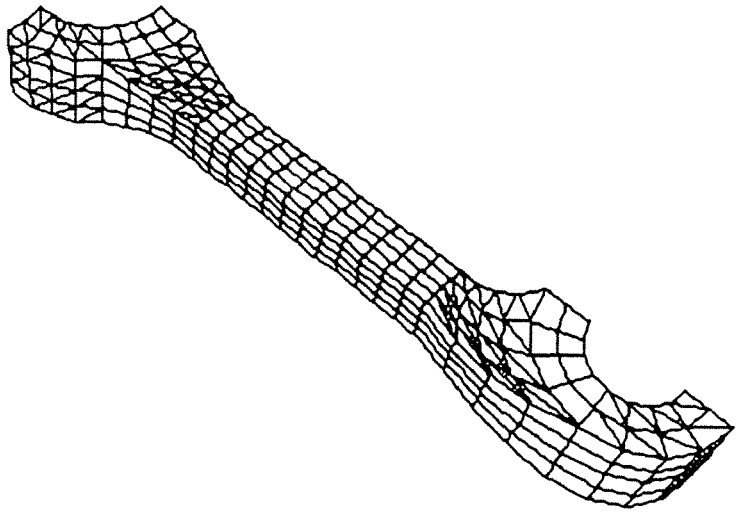


Figure IV-45-b : Le maillage final de la pièce en quadrilatères 8-nœuds (745 nœuds et 448 éléments) avec élimination des parties cachées (environ 30 minutes sur Mac II).

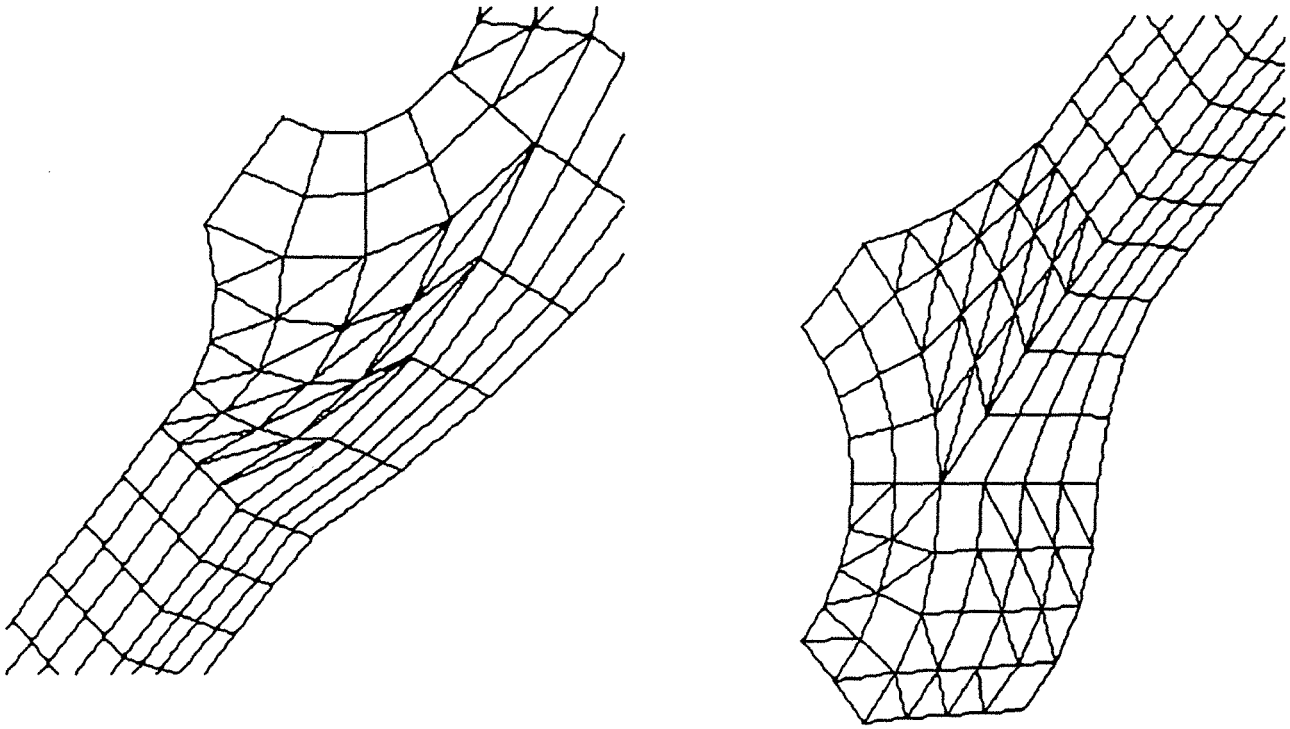


Figure IV-45-c : Un zoom de chacune des extrémités de la pièce

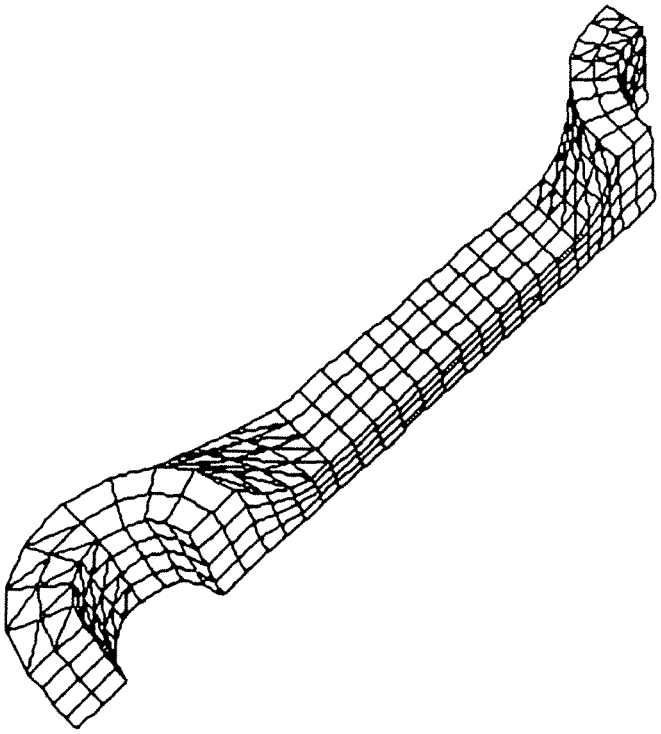


Figure IV-45-d : Même maillage vu d'un autre angle

IV-9 : Conclusions

Nous avons mis au point un logiciel de maillage 3D donnant des résultats acceptables. L'ensemble est géré par des menus déroulants et _ autant que possible _ des saisies directes à l'écran. L'ensemble représente environ 7000 instructions.

Comme nous l'avons vu, ce logiciel présente quelques limites, notamment au niveau du temps de calcul, lorsque l'on souhaite avoir une évaluation des parties cachées. Ceci est le prix à payer pour avoir un ensemble de calculs (tant maillage que visualisation) effectués dans l'espace objet, ce qui permet d'avoir un traitement exact des objets indispensable à la définition de maillage par éléments finis.

Il n'y a cependant aucun doute possible, vu le volume des calculs requis et la complexité des résultats obtenus, les mailleurs automatiques sont encore plus indispensables en 3D qu'en 2D. Les résultats obtenus sont encourageants, mais il reste encore du travail à faire pour obtenir un processus de maillage véritablement automatique et ce, avec des temps de calculs acceptables.

PARTIE B

Résolution par méthodes multigrilles.

Chapitre V

Méthodes Multigrilles

V-1 : Formulation variationnelle et éléments finis

V-1-1 : Introduction

La modélisation des phénomènes physiques est souvent régie par des systèmes d'équations différentielles continues. Dans les cas de géométries simples, de tels problèmes peuvent être résolus d'une manière analytique. Malheureusement, pour les géométries générales, les techniques de déterminations analytiques ne peuvent s'appliquer.

Pour pallier cet inconvénient, de nombreuses méthodes de discrétisation ont été proposées par les mathématiciens ou les physiciens. Elles consistent à décomposer le domaine d'étude en morceaux ou éléments de géométrie simple. La solution du problème sur chaque élément peut être décrite par un nombre fini de paramètres et doit ainsi être facilement accessibles numériquement.

Les premiers types de discrétisations ne furent développés par les physiciens puis les mathématiciens que vers les années 40, essentiellement à partir de considérations pratiques. Elles visaient à traiter certains problèmes par analogie avec des cas simples (par exemple des treillis de barres ou des réseaux de résistances). Vers les années 60, on parvint à concilier les approches empiriques avec les approches variationnelles, permettant de mettre sur pied une méthode générale de discrétisations des problèmes continus.

Le développement de ces méthodes fut aidé par les progrès des ordinateurs, la résolution de problèmes discrets, même de grande taille étant aisé sur ce genre de machine.

L'ensemble des domaines couverts par les méthodes de discrétisation comprend, entre-autres, la mécanique des solides et des fluides, la thermique et l'électricité ; il est donc particulièrement large, d'où son intérêt. Pour la suite de l'exposé, nous nous intéresserons essentiellement à la mécanique du solide.

V-1-2 : formulation

Appelons V, un espace de Hilbert sur $\Re$, de norme $\| \cdot \|$. Cet espace constitue l'ensemble des variations des champs de déplacements possibles.

Les problèmes de mécanique du solide peuvent se résoudre en minimisant l'énergie totale du système. Cette énergie s'exprime souvent sous la forme d'une fonctionnelle

$$J(u)=\frac{1}{2} A(u,u)- f(u) \qquad (1)$$

Où :

_ A(u,v) est une forme bilinéaire continue sur VxV.

_ f(u) est une forme linéaire continue sur V.

On suppose qu'il existe une constante k>0 telle que :

$$\forall\, u,v \in V\ ,\ \| A(u,v) \| \leq k. \| u \|. \| v \|. \quad (2)$$

De plus, cette forme est coercitive, c'est à dire qu'il existe $\alpha > 0$ tel que ;

$$\forall\, u \in V\ ,\ A(u,u) \geq \alpha \| u \|^2 \quad (2')$$

Très souvent, A est en plus symétrique définie positive, c'est à dire que $\forall\, u,v \in V$:

_ $A(u,v) = A(v,u)$

_ $A(u,u) \geq 0$ (3)

_ $A(u,u)=0 \Leftrightarrow u=0$

On montre alors que l'unique minimum de J(u) est solution de l'équation variationnelle :

$$A(u,v)-f(v) = 0 \quad \forall\, v \in V \qquad (4)$$

Pour résoudre ce problème, nous faisons appel à la méthode des éléments finis. Celle ci consiste, comme nous l'avons vu, à découper le domaine de départ en un nombre fini de "morceaux". L'ensemble de ces morceaux constitue un espace vectoriel de dimension $\tilde{v}$ qui est une approximation de V. Il est alors possible d'approcher tout $u \in V$ par $\tilde{u} \in \tilde{V}$.

Si nous désignons par n la dimension de $\tilde{V}$ et par $\{ \phi_1,\ldots,\phi_n \}$ une base, on peut alors écrire :

$$\tilde{u} = \sum_{i=1}^{n} \tilde{u}_i \, \phi_i \quad , \qquad \begin{cases} \tilde{u}_i \in \Re \\ \phi_i \in \tilde{V} \end{cases} \qquad (5)$$

A l'équation variationnelle initiale, on peut associer un problème en dimension finie: trouver l'unique $\tilde{u}$ tel que:

$$\begin{array}{c} A(\tilde{u},\tilde{v}) - f(\tilde{v}) = 0 \\ \text{ou encore} \\ A(\tilde{u},\tilde{v}) = f(\tilde{v}) \end{array} \qquad \forall \tilde{v} \in \tilde{V} \qquad (6)$$

L'explicitation de (5) et de (6) conduit à un système linéaire d'équations que doit vérifier $\{u\} = {}^t\langle u_1,\ldots,u_n \rangle$:

$$[K]\{u\} = \{f\} \qquad (7)$$

avec:

$$k_{ij} = A(\varphi_i,\varphi_j) = k_{ji} \qquad i,j = 1,..,n$$
$$f_i = f(\varphi_i)$$

La matrice [K} obtenue est généralement symétrique définie positive.

V-1-3 : Application à la mécanique des solides

La première phase de l'étude consiste à discrétiser le solide, c'est à dire à découper celui-ci en éléments reliés entre eux par des points appelés "nœuds". Ce découpage doit être compatible avec la notion de maillage par éléments finis vue au chapitre I.

A chaque nœud, sont attachés des variables nodales généralisées (généralement des déplacement et/ou leurs dérivées). Chaque élément est soumis à des forces généralisées (forces et/ou moments) appliquées en ses nœuds.

Chaque élément se comporte exactement de la même façon qu'un morceau de solide de façon que la structure complète discrétisée considérée comme l'assemblage de tous ses éléments suive les mêmes règles que la structure réelle.

Nous allons essayer d'écrire une relation entre forces et déplacements généralisés sous la forme :

$$\{q\} = [K]\{u\} \qquad (9)$$

où : $\{q\}$ est le vecteur forces généralisées.
$\{u\}$ est le vecteur déplacements généralisés.

Pour obtenir cette relation, nous utilisons les approches variationnelles du problème. Parmi ces approches, citons en deux :

_ La première consiste, comme nous l'avons vu, à minimiser l'énergie totale du système.
_ La seconde consiste à utiliser le théorème des travaux virtuels.

Les deux approches sont mathématiquement et physiquement équivalentes. Voyons plus en détail l'approche proposée par la seconde méthode.

V-1-4 : Théorème des travaux virtuels

Considérons un élément soumis en ses nœuds à des forces externes $\{q\}$ et imposons aux nœuds des variations de déplacement $\{\delta u\}$. Cela provoque un travail des forces extérieures

$$\delta W = \langle q \rangle . \{\delta u\} = {}^t\{q\} . \{\delta u\} \qquad (10)$$

Ces forces et déplacements provoquent dans l'élément des champs de déformation et de contraintes σ et $\delta\varepsilon$. L'énergie interne de déformation est alors :

$$\delta U = \int_v \langle \sigma \rangle \{\delta \varepsilon\} \, dv \qquad (11)$$

La condition nécessaire et suffisante pour que le solide soit en équilibre est que l'on ait $\delta W = \delta U$ pour tout variation du champ de déplacement cinématiquement admissible, soit

$$\langle q \rangle \{\delta u\} = \int_v \langle \sigma \rangle \{\delta \varepsilon\} \, dv \quad \forall \, \delta u \in V \qquad (12)$$

V-1-5 : Explicitation formelle

Nous supposons que nous ne formulons notre problème qu'en fonction des déplacements généralisés (formulation simple).

Nous pouvons écrire en tout points de l'élément :

a) approximation nodale : $u = \langle N \rangle \{u\}$ (13)

où <N> est la matrice des fonctions de forme

et {u} est le vecteur des déplacements généralisés aux nœuds.

b) loi de comportement $\{\sigma\} = [D]\{\varepsilon\}$ (14)

c) relation déformation-déplacements

Dans le cas où les déplacement sont petits, les composantes infinitésimales de ε sont données par :

$$\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}) \quad (15)$$

En utilisant a) on peut alors écrire : $\{\varepsilon\}=[B]\{u\}$ (15')

Si l'on prend donc une variation de champ de déplacement cinématiquement admissible $\{\delta u\}$ on obtient finalement en substituant les relations b) et c) dans l'équation des travaux virtuels :

$$\begin{cases} \{\sigma\} = [D][B]\{u\} \\ \{\delta\varepsilon\} = [B]\{\delta u\} \end{cases} \quad (16)$$

: Soit :

$$\langle q \rangle \{\delta u\} = \langle u \rangle \int_v [B]^t [D][B]\{\delta u\}\, dv \quad (17)$$

Ceci devant être valable pour tout $\{\delta u\}$ on obtient donc :

$$\{q\} = \int_v [B]^t [D][B]\, dv \,.\, \{u\} \quad (18)$$

$$\text{soit } \{q\} = [k]\{u\} \quad \text{où} \quad [k] = \int_v [B]^t [D][B]\, dv$$

$[k]$ est une matrice carrée symétrique

V-1-6 : Procédure globale

La structure est discrétisée en éléments finis. Sur chaque élément fini, la matrice de rigidité et le vecteur second membre sont calculée par les relations (19). L'ensemble de ces matrices de rigidité est assemblée en une grande matrice dite "matrice globale". Cette matrice correspond à la modélisation numérique de l'ensemble du la structure conformément à la discrétisation choisie. Elle exprime la relation liant le vecteur global de tous les déplacements généralisés au vecteur global des forces généralisée, cette relation globale s'écrit : $[K_{global}].\{U_{global}\} = \{F_{global}\}$.

Après prise en compte des conditions limites,ce système peut être résolu, donnant ainsi {U}, le vecteur des déplacements recherchés.

On remarquera que la méthode développée dans les paragraphes 1-5 et 1-6 se formalise de la même façon que celle présentée en 1-2.

Remarque

Les problèmes non-linéaires, en particulier ceux relatifs à l'élasticité, conduisent à la même minimisation d'une énergie qui n'est plus quadratique ou même différentiable. On peut cependant associer à cette minimisation des procédures itératives linéaires du même type que celles associées aux problèmes linéaires et qui convergent vers la solution du problème non-linéaire.

V-2 : Méthodes classiques et méthode 2-grilles

V-2-1 : Introduction

La mise en œuvre des méthodes d'éléments finis ou de différence finies conduisent à la constitution puis à la résolution de systèmes linéaires de la forme :

$$[K]\{u\} = \{f\} \qquad (19)$$

Dans la suite de l'exposé, nous nous limiterons au cas des éléments finis. Dès lors, [K] sera la matrice de rigidité de la structure discrétisée, {u} sera le vecteur des déplacement nodaux provoqués par le chargement {f}.

L'un des problèmes essentiels des logiciels d'éléments finis est de résoudre ce système linéaire le plus efficacement possible. Si d'un point de vue théorique, on sait résoudre un tel problème, d'un point de vue numérique, des ennuis peuvent survenir lorsque la taille du système augmente (mauvais conditionnement). Ces ennuis peuvent conduire à l'impossibilité de résoudre numériquement un tel système.

Nous allons donc faire un rapide bilan des méthodes classiques existant en présentant leurs particularités, puis nous présenterons une nouvelle méthode dite "multi-grilles" permettent de pallier aux inconvénients des méthodes classiques.

V-2-2 : Méthodes classiques

Nous diviserons les méthodes classiques en deux catégories : les méthodes directes et les méthodes itératives. Nous allons rappeler les plus connues de ces méthodes sans souci d'être exhaustifs mais plutôt pour en souligner les inconvénients dans le cas de grands systèmes. Pour plus de détail, nous renvoyons le lecteur à [AN 1].

V-2-2-a: Méthodes directes

Les méthodes les plus utilisées sont le pivot de Gauss ou bien les factorisations de matrices (Gauss ou Choleski). Lorsque la matrice est symétrique définie positive (ce qui est souvent le cas), certaines de ces méthodes se simplifient.

Cependant, nous constatons les inconvénients suivants :

- Un nombre de calculs requis devenant rapidement élevé (une estimation asymptotique du nombre d'opérations est la plupart du temps de l'ordre de n^3).
- Une grande sensibilité aux erreurs de chutes entraînant parfois le divergence de la méthode. Ceci est dû au fait que le conditionnement de la matrice tend vers 0 quand l'ordre de la matrice augmente.
- Un grand encombrement mémoire.

A cause de ces inconvénients, on leur préfère les méthodes itératives.

V-2-2-b: Méthodes itératives

On peut distinguer principalement les méthodes de point fixe et les méthodes de gradient.

Les premières consistent à décomposer [K] sous la forme :

$[K] = [M] - [N]$ (20) où [M] est une matrice aisément inversible

On associe alors une suite de vecteur $\{u\}^{(n)}$ tels que :

$$\{u\}^{(n+1)} = [M]^{-1}\,[N]\,\{u\}^{(n)} + [M]^{-1}\,\{f\} \qquad (20)$$

Les choix les plus courants sont :

$$\text{Jacobi relaxé} : [M] = \frac{1}{\omega}[D] \quad \omega \in]0,1[\quad (\text{non relaxé si } \omega=1) \qquad (2'')$$

$$\text{Gauss-Seidel relaxé} : [M] = \frac{1}{\omega}\left([D] + [TI]\right) \quad \omega \in]0,1[\quad (\text{non relaxé si } \omega=1)$$

Avec [D] étant la partie diagonale de [K] et [TI] la partie triangulaire inférieure. La suite de vecteurs $\{u\}^{(n)}$ converge si le rayon spectral $\rho([M]^{-1}[N]) < 1$.
Dans la pratique, ce rayon spectral est souvent proche de 1 par valeur inférieure, ce qui rend la convergence assez lente.

Les deuxièmes méthodes consistent à construire à partir de [K] une fonctionnelle d'énergie :

$$F(\{x\}) = \frac{1}{2}[x]^t[K][x] - [x]^t\{f\}$$

dont on cherchera le minimum par une de gradient conjugué. Le fait que [K] soit symétrique définie positive améliore la convergence et la stabilité des itérations. Malheureusement, la matrice [K] est souvent mal conditionnée, il est donc nécessaire de la préconditionner pour améliorer la convergence.

Parmi les méthodes classiques, les gradients conjugués sont les plus performantes. Cependant, lorsque la taille du système augmente, le nombre d'opérations et donc le temps de calcul croît. De plus, les préconditionnement peuvent être coûteux en place mémoire et ne sont pas toujours stables pour tous les types d'éléments (exemple le préconditionnement ICCG par factorisation incomplète de Choleski).

De façon générale, les méthodes itératives, lorsqu'elles convergent, peuvent être assez lentes et donc pénaliser le temps de calcul. Elles présentent cependant vis à vis des méthodes directes, l'avantage d'être plus sûres car beaucoup moins sensibles aux erreurs de chutes.

V-2-3 : Méthodes 2-grilles

V-2-3-a : Petit historique

Dans les années 40, l'analyse de Fourier a contribué à la découverte d'une des propriétés essentielles des méthodes itératives : l'élimination rapide des hautes fréquences de l'erreur au cours des itérations; propriété appelée lissage.

Vers la même époque, Schröder ou Southwell tentaient de développer des méthodes de calcul utilisant des grilles grossières comme intermédiaire de calcul.

Au début des années 60, Fédorenko pensa à combiner ces deux aspects pour résoudre l'équation de Poisson en une méthode appelée "multigrilles". Il fut suivi par Bachvalof qui étudia les problèmes à coefficients variables. Astrachanof proposa ensuite une formulation multigrilles des problèmes d'éléments finis et démontra le coût optimal de cette méthode.

A partir des années 70, ce furent les Allemands qui reprirent le flambeau de l'étude. En particulier, Brandt conçut une théorie plus générale basée sur les travaux précédents et proposa des méthodes de résolution efficaces pour certains problèmes, essentiellement de mécanique des fluides.

Depuis la fin des années 70, les travaux sur les méthodes multigrilles ont pris un essor considérable, toujours en Allemagne avec Hackbush, Trottenberg, Stuben, ainsi que Maitre en France.

V-2-3-b : Description de la méthodes sur un exemple

1) Préliminaires

L'étude théorique complète des méthodes multigrilles n'a été effectuée que sur quelques problèmes simples, essentiellement en différences finies. En éléments finis, ces études se sont limitées aux problèmes elliptiques. Nous avons donc choisi d'illustrer le comportement fondamental de la méthode multigrilles sur un exemple de différences finies que nous tirons de "Introduction to multigrid methods" de W Hackbush [MG 1].

Définition

La formulation par éléments ou différences finis fait intervenir la notion commune de grille. Par grille, nous entendrons respectivement un maillage par éléments finis ou un maillage pour différences finies. Dans le premier cas, il s'agit d'un maillage conforme au sens des définition vues au chapitre I. Dans le second, il s'agit d'un ensemble de points, autant que possible répartis selon une grille quadrilatérale, et permettant d'approcher les dérivées de tous ordres par des différences.

Nous désignerons par h la taille de la grille. Si la notion de taille est à peu près claire en différences finies, elle est un peu plus flou dans la cas d'un maillage par éléments finis de dimensions très variables. Nous considérerons que la taille de la grille est une quantité pouvant varier localement, et traduisant globalement la qualité avec laquelle la grille permet de modéliser le problème posé. Plus h sera petit, plus précise sera la modélisation, mais en contrepartie, plus complexe sera la résolution.

2) cycle 2-grilles

Nous désirons résoudre le problème de l'équation de Poisson unidimensionnelle avec conditions de Dirichlet :

$$\begin{cases} \Delta (u(x)) = f(x) \quad x \in [0,1] \\ \\ \quad \text{avec } u(0) = u(1) = 0 \end{cases} \tag{21}$$

Nous choisissons une grille de taille h constituée de n+1 points équi-répartis entre 0 et 1 et numérotés selon les x croissants :

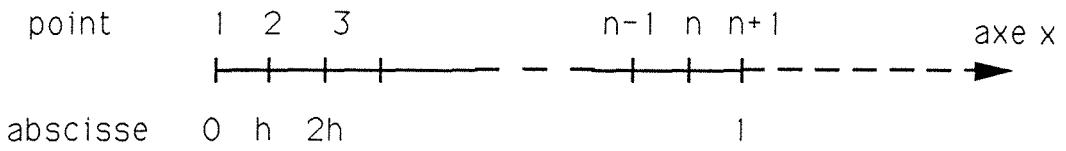


Figure V-1 : Grille utilisée

Nous prenons par ailleurs un schéma de différences finies 3-points:

$$\frac{u(x+h)+u(x-h)-2\,u(x)}{h^2} = \frac{d^2u}{dx^2} + o(h^2) \qquad (21)$$

Le problème ainsi discrétisé permet une formulation matricielle qui, après prise en compte des conditions limites s'écrit :

$$[K_h]\,\{u_h\} = \{f_h\} \qquad (22)$$

avec

$$[K_h] = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & & & & 0 \\ 0 & & & & 1 \\ \vdots & & & & \vdots \\ 0 & & & & 1 \\ 0 & \cdots & 0 & 1 & -2 \end{bmatrix} \quad \{u_h\} = \begin{bmatrix} u(h) \\ \vdots \\ u((n-1)h) \end{bmatrix} \quad \{f_h\} = \begin{bmatrix} f(h) \\ \vdots \\ f((n-1)h) \end{bmatrix}$$

Le cycle 2-grilles se décompose maintenant en deux étapes :

Première étape :

A partir d'une valeur de départ $\{u_0\}$ quelconque, appliquons quelques itérations d'un splitting de Jacobi relaxé (ω=0,5).

On obtient alors l'itérée (i+1) de {u} en fonction de l'itérée (i) par l'équation :

$$\left\{u_h^{i+1}\right\} = \left\{u_h^{i}\right\} + \frac{h^2}{4}\left(\left\{f_h\right\} - \left[L_h\right]\left\{u_h^{i}\right\} \right) \qquad (23)$$

$$= \left[G_h\right]\left\{u_h^{i}\right\} + \frac{h^2}{4}\left\{f_h\right\} \text{ avec } \left[G_h\right] = \left[I_h\right] - \frac{h^2}{4}\left[L_h\right]$$

Si nous notons $\left\{u_h\right\}$ la solution exacte, l'erreur à l'itération i $\left\{e_h^{i}\right\} = \left\{u_h\right\} - \left\{u_h^{i}\right.$

vérifie : $$\left\{e_h^{i+1}\right\} = \left[G_h\right]\left\{e_h^{i}\right\} \qquad (24)$$

Il se trouve alors que $[L_h]$ et $[G_h]$ ont les mêmes vecteurs propres :

$$\left\{V_h^{k}\right\} = \begin{Bmatrix} \sin(k\pi h) \\ \sin(2k\pi h) \\ | \\ \sin((n-1)k\pi h) \end{Bmatrix} \quad k=1,..,n-1 \qquad (24')$$

et que les valeurs propres de $\left[G_h\right]$ sont les $s_k = \cos^2\left(\frac{k\pi h}{2}\right)$ $k=1,..,n-1$.

Dès lors, le taux de convergence de la méthode de Jacobi relaxé (ω=0,5) est son rayon spectral $\rho = \sup_k |s_k|$ soit $\rho = \cos\left(\frac{\pi h}{2}\right) = 1 - \frac{(\pi h)^2}{8} + o\left(h^2\right)$.

Ce rayon est très voisin de 1, ce qui explique la lente décroissance de l'erreur. Par contre, si l'on se limite aux hautes fréquences de l'erreur (k>(n-1)/2) , le taux de convergence devient très voisin de zéro, ce qui signifie que ces hautes fréquences sont très rapidement détruites.

<u>En conclusion de cette étape</u>

Après quelques itérations de l'opérateur $[G_h]$, l'erreur globale n'a pas notablement décru, mais elle a acquis un aspect régulier semblable à ce que montre la figure V-2 ci-dessous. Cet aspect fondamental s'appelle le <u>lissage</u> et est commun à la quasi totalité des processus itératifs.

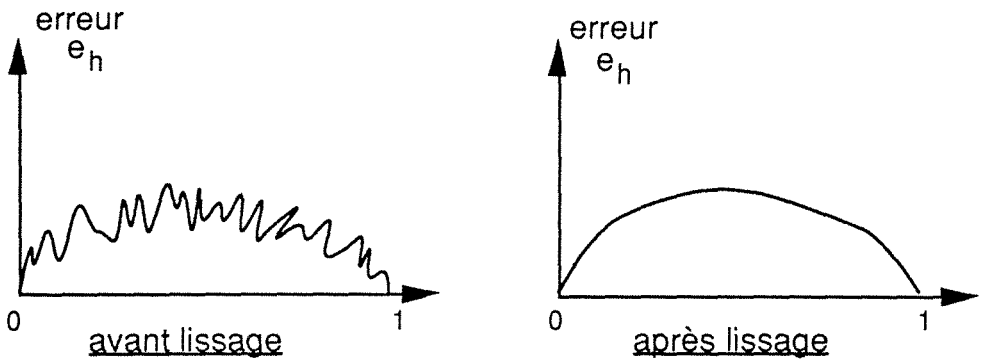


Figure V-2 : propriétés de lissage

Seconde étape

Après quelques itérations de lissage, désignons par :

* $\{u_h\}$ la solution exacte

* $\{\tilde{u}_h\}$ la solution approchée obtenue

* $\{e_h\} = \{\tilde{u}_h\} - \{u_h\}$ l'erreur sur $\{u\}$

* $\{d_h\} = [L_h]\{u_h\} - \{f_h\}$ le défaut

On peut remarquer que par linéarité, nous avons :

$$\{d_h\} = [L_h]\{u_h\} - \{f_h\} = [L_h]\{e_h\} \quad (25)$$

Nous retombons sur un système tout à fait analogue au système de départ et faisant intervenir la même matrice $[L_h]$. Ce système définit ce qu'on appelle l'équation de correction. Si l'on sait résoudre cette équation, soit $\{e_h\} = [L_h]^{-1}\{u_h\}$, on en déduit :

$$\{u_h\} = \{\tilde{u}_h\} - \{e_h\} \quad (26)$$

Cependant, ce problème est aussi complexe à résoudre que le problème de base. On va donc se ramener sur une grille plus grossière de taille H=2h, grille que nous désignerons par Ω_{2h}. Pour passer Ω_h de à Ω_{2h}, nous définissons une restriction r (il n'y a bien sur pas unicité) :

$$r : \Omega_h \rightarrow \Omega_{2h}$$
$$\{d_h\} \rightarrow \{d_{2h}\}$$

$$\text{tel que} \quad d_{2h}(x) = \frac{1}{4}\left[d_h(x-h) + d_h(x+h) + 2d_h(x)\right] \qquad (27)$$

$$\text{le système } \left[L_h\right]\left\{e_h\right\} = \left\{d_h\right\} \text{ est remplacé par } \left[L_{2h}\right]\left\{e_{2h}\right\} = \left\{d_{2h}\right\} \qquad (28)$$

Ce système est de taille plus réduite que le premier, il est donc plus aisé à résoudre par quelque méthode que ce soit.

$$\text{nous obtenons alors : } \left\{e_{2h}\right\} = \left[L_{2h}\right]^{-1}\left\{d_{2h}\right\} \qquad (29)$$

que allons alors prolonger sur la grille Ω_h pour avoir une estimation de $\{e_h\}$ par un opérateur de prolongation p :

$$\begin{aligned} p : \Omega_{2h} &\rightarrow \Omega_h \\ e_{2h} &\rightarrow \widetilde{e}_h \end{aligned}$$

$$\text{tel que } \widetilde{e}_h(x) = \begin{cases} e_{2h}(x) \text{ si } x \text{ appartient à la grille grossière } \Omega_{2h} \\ \\ \dfrac{1}{2}\left(e_{2h}(x-h) + e_{2h}(x+h) \right) \text{ sinon} \end{cases} \qquad (30)$$

Nous obtenons ainsi une nouvelle valeur approchée de la solution comme étant :

$$\widehat{u}_h = \widetilde{u}_h - p\left(e_{2h}\right) = \widetilde{u}_h - \widetilde{e}_h \qquad (31)$$

Résumons le principe de l'algorithme 2-grilles par le schéma suivant :

1) lissage

on effectue v1 itérations de lissage : $\{\tilde{u}\} = \mathbf{S}_h^{v1}(\{\tilde{u}\})$

2) correction sur grille grossière

calcul du défaut $\{d_h\} = [L_h]\{\tilde{u}_h\} - \{f_h\}$

restriction du défaut $\{d_{2h}\} = r(\{d_h\})$

résolution de l'équation de correction $\{e_{2h}\} = [L_{2h}]^{-1}\{d_{2h}\}$

prolongement $\{\tilde{e}_h\} = p(\{e_{2h}\})$

correction de la solution $\{\hat{u}_h\} = \{\tilde{u}_h\} - \{\tilde{e}_h\}$

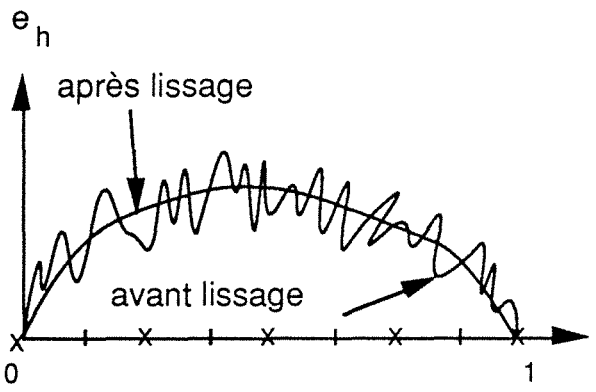
3) post-lissage

on effectue éventuellement v2 itérations : $\{\hat{u}\} = \mathbf{S}_h^{v2}(\{\hat{u}\})$

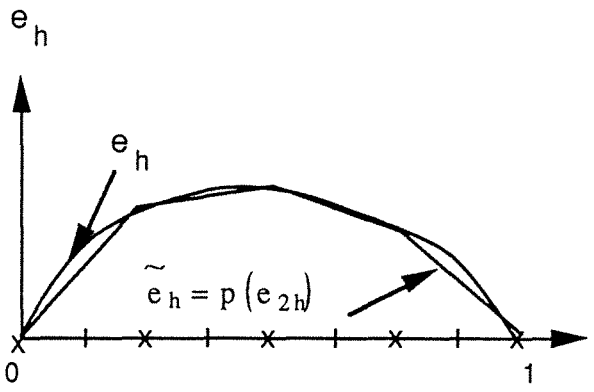
Tableau V-1 : Cycle 2-grilles

L'efficacité de la méthode réside dans l'association de deux méthodes plutôt mauvaises. Le lissage - processus lent à converger - sert à donner à l'erreur un aspect régulier. Dès lors, la correction sur grille grossière - processus généralement divergent - assure une correction sur grille de la solution convergente. La conjugaison judicieuse des propriétés de ces deux méthodes nous donne une bonne méthode.

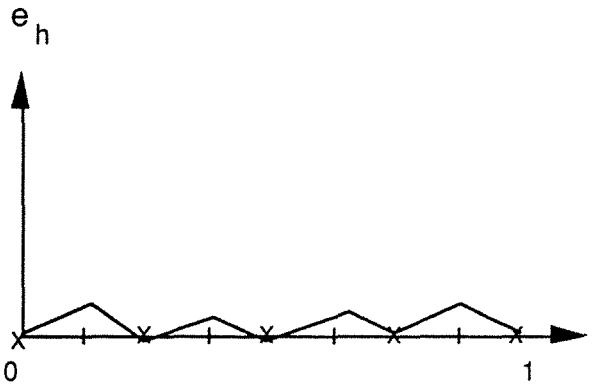
L'ensemble du comportement de la méthode peut se résumer par la figure V-3 ci-dessous :



destruction des hautes fréquences de l'erreur par lissage



après résolution de l'équation de correction, on peut efficacement approcher e_h par $p(e_{2h})$



x = points de la grille grossière
I = points communs aux 2 grilles

après correction ,les basses fréquences de l'erreur ont été supprimées . Sur les deux étapes, l'erreur a fortement décru

Figure V-3 : comportement de l'erreur e_h sur un cycle 2-grilles

On a l'habitude de représenter les composantes du cycle 2-grilles par le dessin suivant :

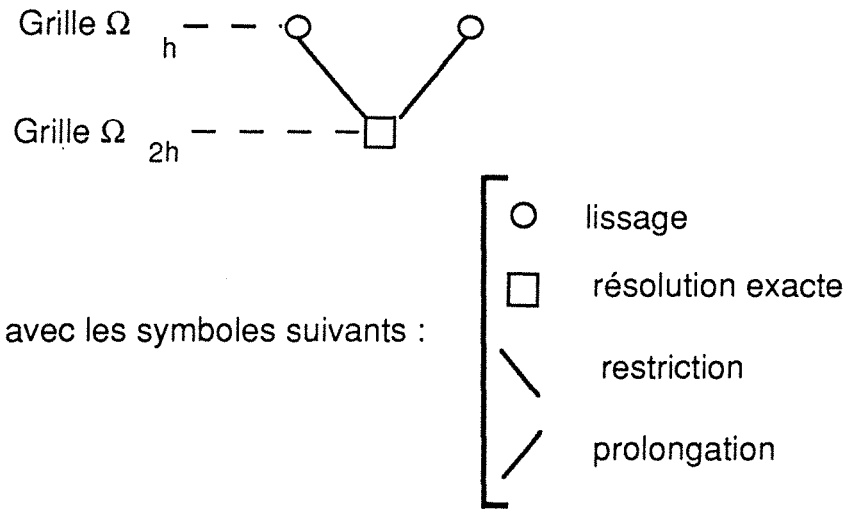


Figure V-4 : schématisation d'un cycle 2-grilles

V-3 : Méthodes multigrilles

Nous sommes toujours dans le cas du problème régi par les équations (21).Bien souvent, dans la résolution de systèmes de grande taille, un simple cycle 2-grilles ne suffit pas. Certes, sur la grille grossière, la taille du problème à résoudre est moindre, mais il se peut que cela soit encore trop pour l'ordinateur ou l'utilisateur.

Maintenant, on peut remarquer que, parti du système $[L_h]\ \{u_h\} = \{f_h\}$, le cycle 2-grilles nous conduit à résoudre un autre système $[L_{2h}]\ \{u_{2h}\} = \{f_{2h}\}$. L'idée de base va donc être de résoudre par un processus 2-grilles et ainsi de suite.

On va donc traiter chaque nouveau système apparaissant de façon récursive jusqu'à ce que l'on arrive à un niveau de grille où le problème sera de taille suffisamment faible pour qu'on puisse le résoudre : on obtient ainsi la méthode multigrilles.

Dans la pratique, pour notre problème, nous allons définir une grille de départ Ω_0 (la plus grossière) et des grilles $\Omega_1,\ldots,\Omega_n$ par exemple obtenu chacune par dichotomie de la précédente (ie $h_k=2h_{k+1}$). Pour chaque changement de grille, nous utiliserons les prolongations et restrictions vues au paragraphe V-2-3-b.

Nous désignerons par :

$[L_k]$: la matrice au niveau k
S_k : l'opérateur de lissage
r_k : la restriction de Ω_k vers Ω_{k-1}
p_k : la prolongation de Ω_{k-1} vers Ω_k

On peut alors représenter un pas de processus multigrilles lancé au niveau k pour résoudre le système $[L_k]\{u_k\} = \{f_k\}$ par l'algorithme suivant. Comme au paragraphe V-2-3-b, le processus se décompose en 3 phases: pré-lissage-correction-post-lissage. Nous noterons respectivement $\{u_k^{1/3}\}$, $\{u_k^{2/3}\}$, $\{u_k^{1}\}$ les résultats de ces trois phases. Nous noterons $\{u_k^{0}\}$ la valeur de départ de $\{u_k\}$ dans l'itération.

si k = 1	u_k^1 est donné par l'algorithme 2-grilles entre les grilles Ω_0 et Ω_0
si k ≥ 1	a) lissage $\left\{u_k^{1/3}\right\} = S_k^{\nu}\left(k, \{f_k\}, \left\{u_k^{0}\right\}\right)$ b) correction _ calcul du défaut $\{d_k\} = [L_k]\left\{u_k^{1/3}\right\} - \{f_k\}$ _ restriction du défaut : $\{d_{k-1}\} = r(\{d_k\})$ _ on obtient l'équation de correction $[L_{k-1}]\{e_{k-1}\} = \{d_{k-1}\}$ qui est résolue en γ_{k-1} pas de la méthode k-grilles sur $(\Omega_{k-1},\ldots,\Omega_0)$ en prenant le vecteur nul comme valeur de départ pour $\{e_{k-1}\}$ _ $\left\{u_k^{2/3}\right\} = \left\{u_k^{1/3}\right\} - p\left(\{\tilde{e}_{k-1}\}\right)$ c) post-lissage $\left\{u_k^{1}\right\} = S_k^{\nu 2}\left(k, \left\{u_k^{2/3}\right\}, \{f_k\}\right)$

Tableau V-2 : cycle multigrilles

Dans la pratique, γ_K est généralement constant. On se limite couramment à γ_K=1 (V-cycle) ou γ_K=2 (W-cycle) pour des raisons que nous verrons plus loin. Ces deux types de cycle peuvent s'illustrer sur l'exemple suivant comportant 3 grilles (figure V-5).

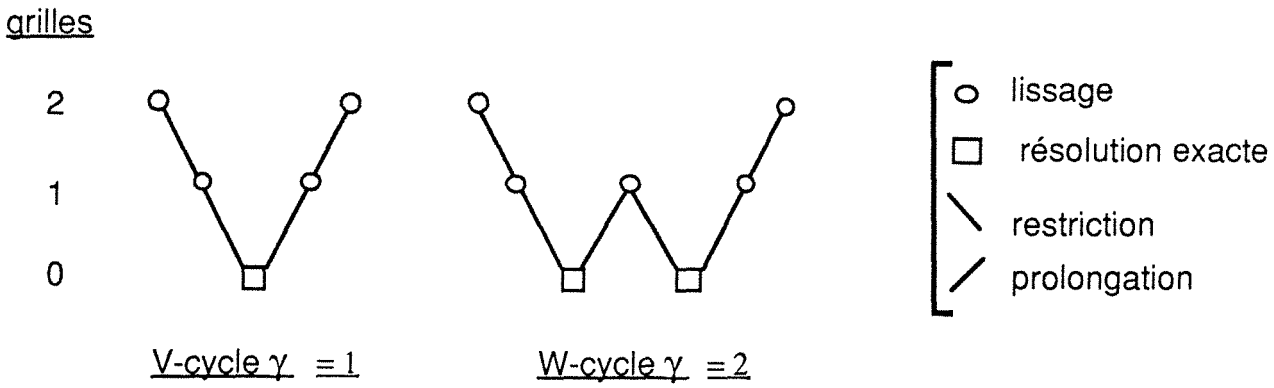


Figure V-5 : V- et W-cycles

On peut reprendre ces illustrations sous une forme plus informatique :

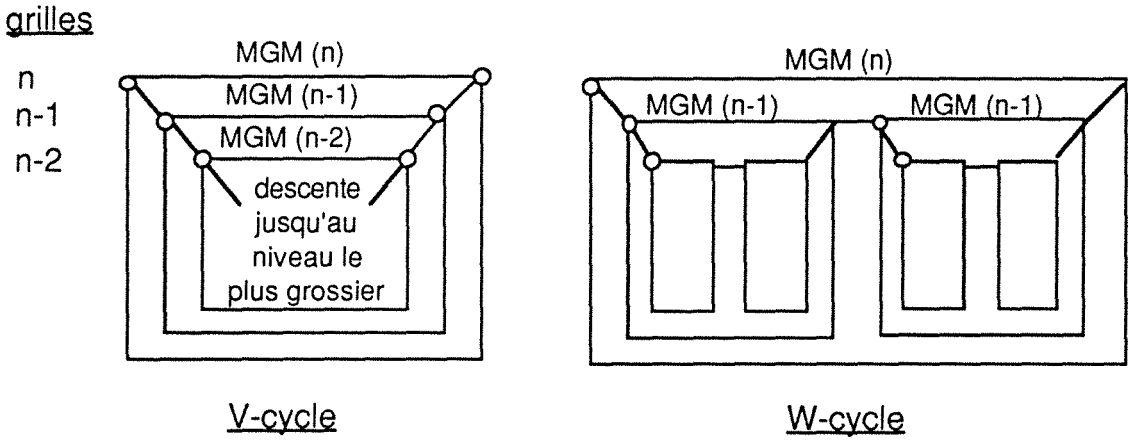


Figure V-6 : V- et W-cycles

V-4 : Convergence de la méthode multigrilles

V-4-1 : Etude générale

Une étude globale exacte de la convergence de la méthode est extrêmement difficile - voire impossible - de par les nombreux paramètres qui la composent. Ce chapitre constitue aujourd'hui encore un axe important de recherches.

L'analyse exacte du cycle 2-grilles a été réalisé par Trottenberg et Stüben sur certains problèmes particuliers en différences finies, notamment :

$$\left\{ \begin{array}{c} -\Delta(u) = f \\ -\Delta(u) \pm a\ u = f \\ \Delta(\Delta(u)) = \phi \end{array} \right\}$$

Avec en outre certaines conditions à vérifier :

- _ Domaines rectangulaires
- _ opérateurs L_h à coefficients constants
- _ conditions limites symétriques
- _ L_h symétrique

Cette étude est possible car dans ce cas, on connait les fonctions propres des opérateurs intervenant, on va alors faire une analyse de Fourier globale du cycle. Pour les cas plus complexes, on est obligé de recourir à l'analyse de Fourier locale proposée par Brandt [MG 1] [MG 2].

L'idée commune aux deux types d'analyses de Fourier est d'évaluer (de façon exacte ou approchée) le rayon spectral ρ du cycle. A partir de la définition d'un lisseur S_h, on est amené à définir un facteur de lissage :

$$\bar{\mu}(\omega) = \sup_h \left(\sup_{\substack{\text{hautes} \\ \text{fréquences}}} |a(h,\omega)| \right) \quad (31)$$

où :

h est la taille de la grille

A(h,ω) sonr les valeurs propres de l'opérateur S_h

ω est le facteur de sur-relaxation

Il s'agit là d'une valeur extrémale du facteur de lissage prise sur chaque grille. Cette valeur rend compte du lissage des hautes fréquences. A partir de là, une étude complète nécessiterait de s'intéresser à r, p , $[L_{2h}]^{-1}$. Brandt, quant à lui , pense que l'étude du seul $\bar{\mu}$, suffit à donner une approximation de la convergence. La pratique a monté que cette analyse était d'une surprenante précision. Malheureusement, cette étude est difficile à mener dans le cadre d'un problème quelconque d'éléments finis dans la mesure où la notion de hautes et basses fréquences dépend de la façon géométrique dont on passe d'une grille à une autre.

V-4-2 : Autre preuve de convergence

Hackbush [MG 1] [MG 2] propose une preuve de convergence qui a l'intérêt de s'appliquer à presque tous les cas (différences ou éléments finis).

Le cas du cycle 2-grilles

le processus d'un cycle 2-grilles peut se mettre sous la forme d'un seul opérateur M_h s'écrivant :

$$M_h = \left(I - p\, L_H^{-1}\, r\, L_h \right) S_h^{\nu} \qquad (32)$$

où

- I représente l'identité
- Ω_h et Ω_H sont respectivement les grilles fine et grossière
- h et H sont les tailles respectives de grilles Ω_h et Ω_H
- p est la prolongation de Ω_H vers Ω_h
- r est la restriction de Ω_h vers Ω_H
- L_h et L_H sont les opérateurs décrivant le problème sur les grilles Ω_h et Ω_H respectivement
- S_h^{ν} est l'opérateur décrivant les ν itérations de lissage

On peut alors écrire :

$$\| M_h \| \leq \| L_h^{-1} - p\, L_H^{-1}\, r \| . \| L_h\, S_h^{\nu} \| \qquad (33)$$

La convergence sera assurée si $\| M_h \| \leq 1$.

L'écriture de $\| M_h \|$ sous forme de produit permet d'analyser séparément les deux composantes essentielles de la méthode 2-grilles :

- le lissage
- la correction sur grille grossière

* S_h^ν sera réputé avoir une "propriété de lissage" s'il existe $c(\nu)$, $\bar{\nu}(h)$ et α tels que:

$$\| L_h S_h^\nu \| \leq c(\nu)\, h^{-\alpha} \quad \text{pour } 1 \leq \nu \leq \bar{\nu}(h) \qquad (34)$$

$$\text{avec} \begin{cases} c(\nu) \to 0 \\ \nu \to \infty \\ \text{et} \\ \bar{\nu}(h) \to \infty \\ h \to 0 \end{cases}$$

Très souvent, c et ν, $\bar{}$ sont indépendants de h et l'on peut écrire:

$$\| L_h S_h^\nu \| \leq c(\nu)\, h^{-\alpha} \quad \text{pour } \nu \leq 1 \qquad (34')$$

* On obtient une "propriété d'approximation" si l'on peut écrire :

$$\| L_h^{-1} - p\, L_H^{-1}\, r \| \leq C_1\, h^{\alpha} \qquad (35)$$

avec α étant la valeur décrite en 34 et 34'
et C_1 étant une constante positive

Cette propriété traduit le fait que la solution sur la grille grossière approche correctement l'inconnue recherchée.

Dans la pratique, il suffit que ces deux propriétés ci-dessus soient vérifiées et qu'il existe h_{max} et ν_{min} tels que :

$$c(\nu) < 1/C_1 \quad \text{pour } \nu \geq \nu_{min} \text{ et } h \leq h_{max}$$

pour que $\| M_h \| \leq 1$ et donc que le processus 2-grilles converge.

Le cas du cycle multigrilles

Si l'on admet que les deux propriétés (34 et 35) sont vérifiées, Hackbush [MG 2] montre qu'avec quelques propriétés supplémentaires sur r , sur p et sur l'emboîtement des grilles successives, la méthode multigrilles converge à tous niveaux pour $\nu \geq \nu_{min}$, $h \leq h_{max}$ et $\gamma \geq 2$.

Cette démonstration présente le grand avantage de montrer un des aspects essentiels de la méthode multigrilles : un taux de convergence indépendant de h. Cependant, la globalité de la méthode a pour inconvénient de ne pas toujours rendre compte des particularités de chaque problème, d'autant plus que la preuve de convergence ne donne qu'une valeur asymptotique parfois éloignée de la valeur réelle ; enfin , elle ne porte que sur le calcul de la norme.

V-5 : Etude des paramètres de la méthode

Nous allons maintenant détailler les diverses opérations constituant l'ensemble de la méthode multigrilles pour en voir les particularités. Comme nous allons le voir, ces divers composantes ne sont pas indépendantes les uns des autres.

V-5-1 : Lissage

On trouve de nombreux types de lisseurs, essentiellement parmi les méthodes itératives. On peut citer :

_ Les processus de relaxation (Gauss, Jacobi avec ordre de points, par blocs, par lignes ou par colonnes)

_ Les gradients

_ Les méthodes de décomposition Lower-Upper complètes ou incomplète.

L'efficacité des lisseurs peut se mesurer différemment. Hackbush [MG 2] définit un facteur de lissage ("smoothing number") valant :

$$Q_L = \sup_{\Omega_h} \left(\| L_h S_h^{\nu} \| \Big/ \| L_h \| \right) \qquad (36)$$

Ce facteur joue un rôle analogue à celui du taux de lissage μ défini par Brandt [MG 1] [MG 2] et vu au paragraphe V-4.

Ces deux valeurs servent à approcher ρ^* , le taux de convergence (généralement inconnu) de la méthode multigrilles. μ est généralement plus proche de ρ^* que Q_L pour les faibles valeurs de ν (1 à 3) alors que c'est l'inverse qui se produit pour des valeurs de ν plus élevées.

Il est possible dans certains cas de différences finies, d'améliorer l'efficacité des lisseurs par une numérotation judicieuse des points, mais cela est beaucoup plus difficile à faire dans le cas des éléments finis, d'autant plus que d'autres problèmes de numérotation des points interviennent (largeur de bande par exemple).

D'une manière plus pratique, nous avons testé les 3 processus de lissage suivants :

_ 1 Jacobi relaxé (de facteur de sur-relaxation $\omega \in]0,2[$)

_ 2 Gauss-Seidel relaxé (de facteur de sur-relaxation $\omega \in]0,2[$)

_ 3 Gradient conjugué

La troisième méthode est la plus efficace mais nécessite plus de calculs, nous nous limiterons donc aux deux premières méthodes , la seconde étant la plus performante des deux. Le choix du paramètre ω sera discuté plus loin. Quelques itérations d'une de ces deux méthodes (entre 3 et 5) suffisent généralement.

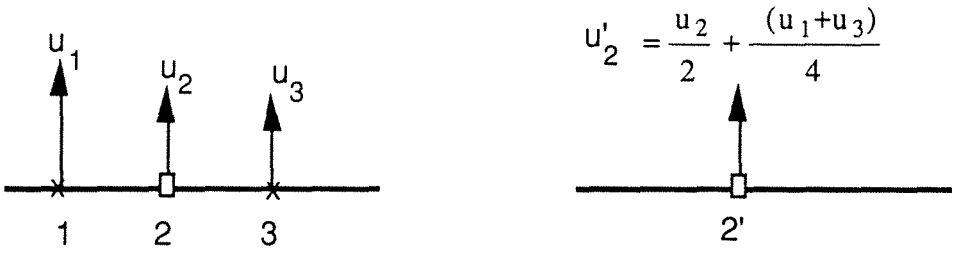
Hackbush [MG 2] propose quelques méthode apparemment efficaces, basées sur des factorisations de matrice. Nous les avons exclues pour la complexité de leurs calculs et leur place mémoire requise.

Le choix du lisseur est l'un des points principaux des méthodes multigrilles. Il s'agit de lisser efficacement et à moindres frais l'erreur pour être sûr que le processus peut que le processus peut converger. C'est cette idée qui a présidé à notre choix. Dans un cas plus particulier (problème mal conditionné), Maitre [MG 4] estime que les lisseurs sont à chercher parmi les méthodes que l'on aurait utilisées comme solveurs.

V-5-2 : Restrictions et prolongations

Deuxième élément-clé des méthodes multigrilles, ces opérateurs permettent de transmettre l'information de grille en grille.

Dans le cas des restrictions, on procède généralement par moyennes pondérées. Un point commun aux deux grilles (fine et grossière) se voit affecté sur la grille grossière une valeur qui est la moyenne entre sa valeur et celles de ses voisins, toutes prises sur la grille fine. On trouvera figure V-7 un exemple trivial de restriction unidimensionnelle.



□ : points communs aux 2 grilles

x : points de la grille fine

Figure V-7 : Exemple de restriction

Notons un cas particulier de restriction dite "restriction injective triviale" consistant à prendre sur la grille grossière la valeur qu'avait le point sur la grille fine. Cela économise pas mal de calculs mais cela peut provoquer de fortes instabilités lors de la correction sur grille grossière; nous l'avons donc exclue.

Les prolongations suivent un processus similaire. Les valeurs manquantes sont générées par interpolation comme le montre l'exemple ci-dessous (Figure V-8).

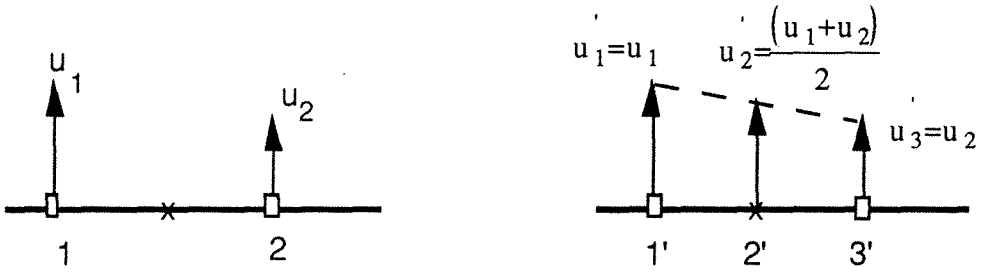


Figure V-8 : Un exemple de prolongation

Il est particulièrement astucieux de prendre :

$$r = p^t$$

Cela "symétrise" le processus et permet, notamment dans le cadre des éléments finis, d'avoir une formulation plus simple.

V-5-3 : Matrices de rigidité

Partis du problème à résoudre $[L_h]\ \{u_h\} = \{f_h\}$, nous nous ramenons à $[L_H]\ \{u_H\} = \{f_H\}$. La question de pose donc de savoir que prendre pour $[L_H]$. Deux options sont possibles :

- _ Utiliser à tous niveaux le même schéma de discrétisation qu'au niveau le plus fin, c'est à dire, en éléments finis, recalculer à chaque niveau la matrice de rigidité.
- _ Utiliser l'approche de Galerkin [MG 1] [MG 2] consistant à prendre $[L_H] = r\ [L_h]\ p$.

La première approche présente les avantages suivants

- _ La définition de [L] à tous niveaux est indépendante des niveaux précédents et suivants.
- _ Essentiellement en différences finies, une simplification des calculs.

En outre, l'approche de Galerkin présente les inconvénients suivants :

- _ La définition de toutes les matrices $[L_H]$ dépend de $[L_h]$
- _ Dans le cas d'un processus full-multigrilles (que nous verrons plus loin), l'opérateur $[L_H]$ utilisé dans les phases de montée et l'opérateur $[L_H]$ utilisé dans le processus multigrilles ne sont pas les mêmes alors qu'ils traduisent le même problème sur la même grille Ω_H

En contrepartie, cette approche présente les avantages suivants :

- _ Si $[L_h]$ est symétrique (définie positive), on peut avoir $[L_H]$ symétrique (définie positive) si $r = p^t$. Ceci est particulièrement avantageux dans le cas des éléments finis comme nous l'avons mentionné plus haut.
- _ Cela permet une meilleure vérification de la propriété d'approximation dans certains cas dépendant essentiellement de l'ordre de r et p.

En fait, la principale raison conduisant au choix de l'approche réside dans les problèmes de stockage et de temps de calcul. Selon les contraintes que l'on s'est imposé, on doit choisir l'approche convenant le mieux à son programme. En particulier, le stockage de la topologie de chaque matrice ainsi que l'aspect "creux" de cette dernière sont des éléments décisifs pour le choix de l'approche. Ces deux aspects sont déjà importants dans un logiciel classique d'éléments finis, ils sont encore cruciaux lorsque s'y ajoute un processus multigrilles (avec plein de maillages et donc de matrices).

V-5-4 : Types de cycles

Le nombre de corrections sur grille grossière permet de définir le type de cycle (V- , W-cycles ou autres).

Il est cependant évident que plus γ est élevé, mieux la méthode converge. Cependant, cette convergence n'est pas uniforme. Au fur et à mesure que γ augmente, un effet de "saturation" intervient dans le taux de convergence.

Pour illustrer ce propos, nous avons étudié une poutre de section carrée, encastrée à une extrémité et soumise en son autre extrémité à un couple de torsion d'axe 0x (figure V-9).

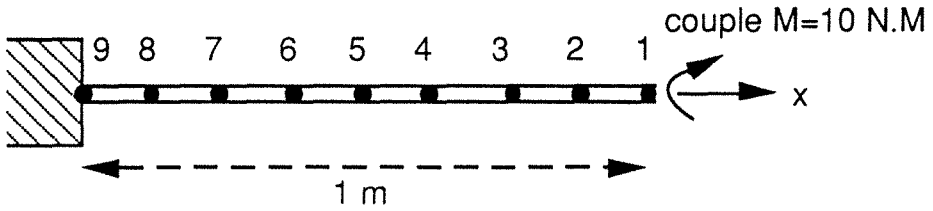
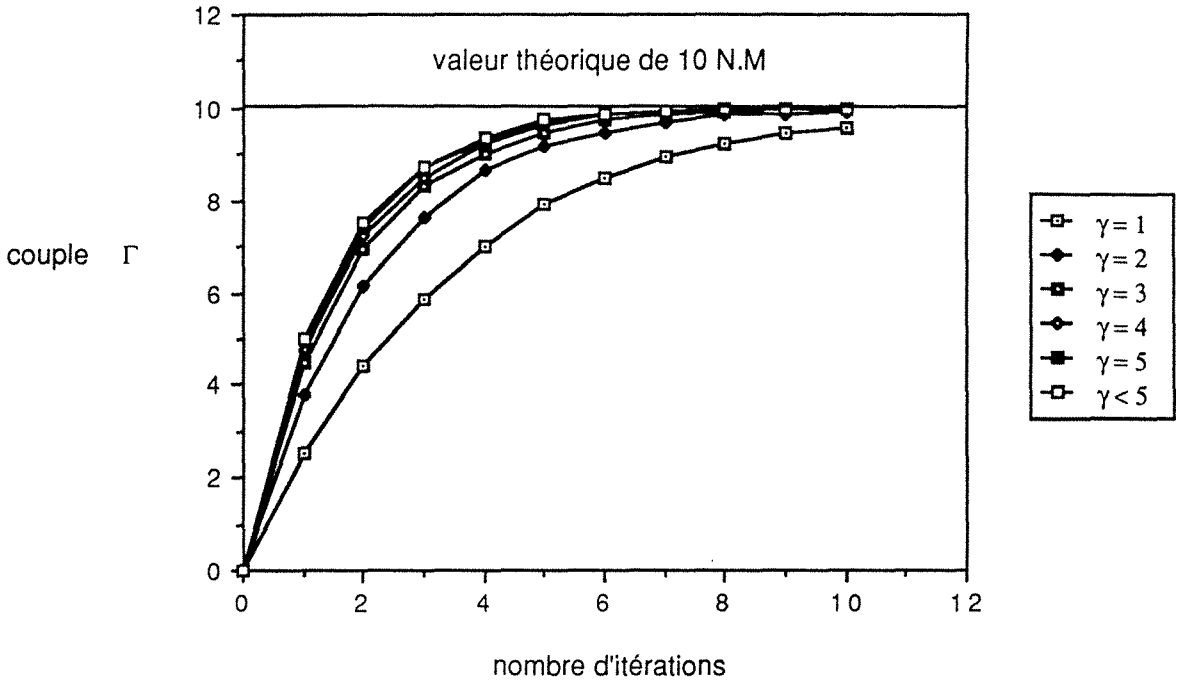


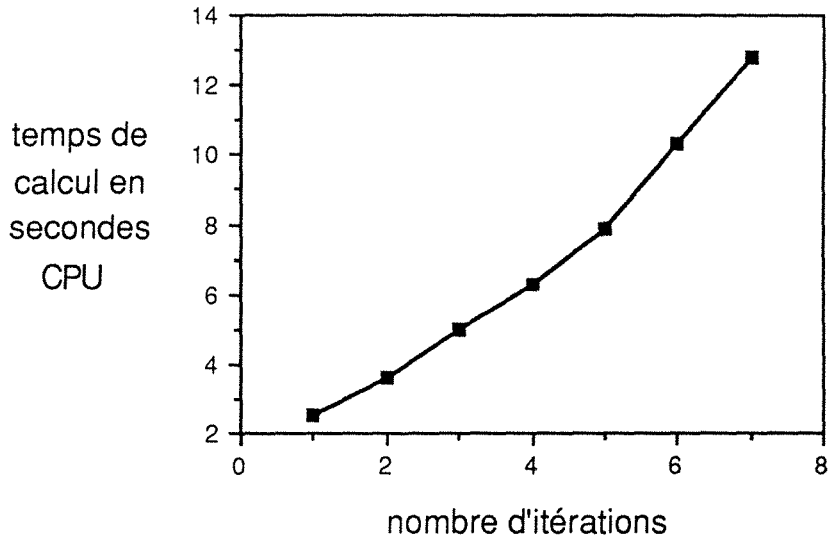
figure V-9 : Dessin de la grille la plus fine

Nous avons utilisé une méthode 3-grilles en faisant varier de γ 1 à 10. Le maillage le plus grossier se compose des nœuds 9, 5 et 1. Les deux autres grilles sont obtenues par dichotomies successives pour obtenir la figure V-10. Le maillage de la grille est constitué d'éléments 2-nœuds.

Notre valeur test est le couple à l'encastrement Γ valant théoriquement 10 N.M. Nous avons tracé la valeur Γ en fonction du nombre d'itérations pour γ variant de 1 à 7. En parallèle, nous traçons le temps de calcul en fonction de γ. Cet essai a été réalisé en 1986 sur un ordinateur DPS 6.



Courbe V-1:Valeur numérique de Γ en fonction du nombre d'itérations et de γ



Courbe V-2:temps de calcul (en secondes) en fonction de γ

Comme on peut le constater, le gain convergence décroît avec γ. A l'opposé, le temps de calcul croit avec γ. Hackbush [MG 1] estime que le simple passage de γ=1 à γ=2 provoque environ 50 % de calculs en plus. Nos essais sur le temps de calcul donnent pour chaque augmentation de γ de 1 ,un surcoût d'environ 25 à 50 % .

Il ressort de cela qu'une valeur de γ entre 1 et 3 est raisonnable et, qu'en général, γ=2 offre un bon compromis rapidité-précision.

V-5-5 : Successions de grilles

En général, la succession de grilles est effectuée par division successives d'une grille de départ (la plus grossière). Ces divisions se font généralement de façon homogène. En différences finies, on rajoute des points intermédiaires de diverses façon possibles (par ajout de points milieux suivant toutes les directions, suivant une direction privilégiée, ajout de points en échiquier). En éléments finis, on procède par divisions d'éléments, le tout étant que le maillage subdivisé soit correct au sens des éléments finis.

Il est aussi possible de diviser sélectivement surtout en éléments finis. Cela permet de raffiner sélectivement le maillage là où cela est jugé nécessaire (nous reviendrons plus longuement au chapitre 8 sur la façon de faire et sur les critères de division). Il est à noter qu'un tel type de technique est aussi possible en différences finies, mais il est plus délicat à mettre en œuvre [MG 1].

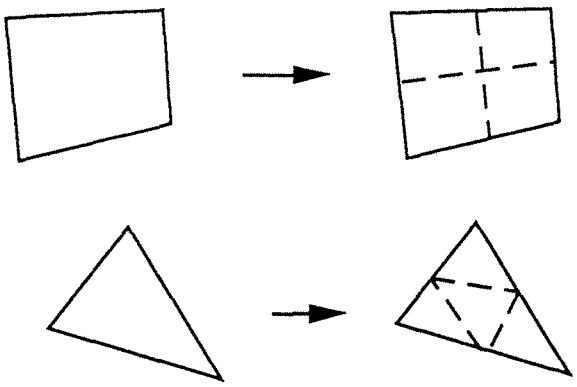


Figure V-10 : Exemple de divisions de maillage

Ces opérations de divisions nécessitent tout un processus de redéfinition des maillages et ce pour chaque grille (ie topologie des éléments, coordonnées des nœuds, matériaux, description des frontières, ...). Il s'agit là d'un processus assez pénible mais indispensable. Au niveau du 1D, cela reste encore assez aisé, par contre, en 2D ou 3D, le volume des calculs croît assez vite.

De façon générale, ces opérations de redéfinition de maillage ainsi que les transferts de variables de grilles en grilles (prolongations et restrictions) et le calcul-stockage des diverses matrices de rigidité pénalisent assez fortement le temps d'éxécution. Une étude préalable doit donc être menée pour savoir quelles seront d'une part, les structures de données et d'autre part, les méthodes de calculs les plus judicieuses à utiliser. Ce sera d'ailleurs l'objet de notre prochain chapitre (A suivre ...).

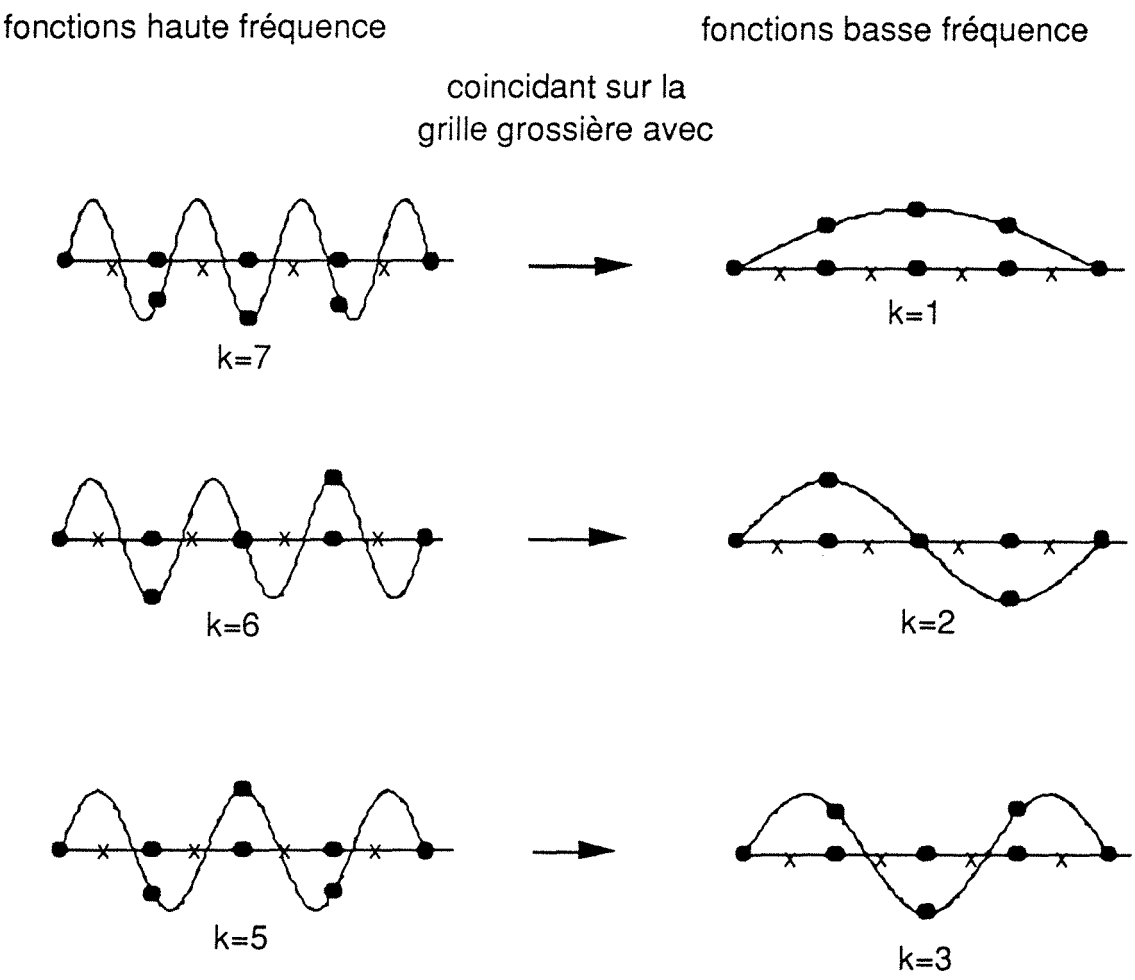
V-5-6 : Problèmes de coordination entre lissage et correction

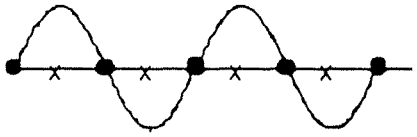
Sur l'exemple du cycle 2-grilles présenté en début de chapitre, nous avons arbitrairement effectué une partition dans l'ensemble des N fonctions propres de l'erreur entre les "hautes" et les "basses" fréquences. Dans la pratique, ce distingo ne peut être fait de façon générale car il dépend du type de changement de grille effectué.

Pour illustrer ce propos, prenons l'exemple du cycle 2-grilles. Parti d'un vecteur de départ $\{u_0\}$ quelconque, au début du cycle, le vecteur erreur $\{e\}$ possède un certain nombre de fonctions propres sur la grille fine. Le nombre de ces fonctions propres dépend de la taille du maillage. Lorsque l'on passe sur la grille grossière, le nombre de points diminue. Dès lors, il peut ne plus y avoir assez de points pour décrire correctement les variations de toutes les fonctions. Ce sont bien évidemment les plus hautes fréquences qui pâtissent le plus de ceci, les plus basses (ie les fonctions les plus lisses) n'ayant besoin que de peu de

points pour être décrites correctement. Il y a donc "disparition" de certaines fonctions. Nous appelerons donc désormais "basses fréquences" les fonctions propres subsistant sur la grille grossière et "hautes fréquences" les autres fonctions.

On trouvera ci-dessous un exemple des fonctions propres du problème (21) utilisant la modélisation (22) (équation de Poisson unidimentionnelle en différences finies).





k=4

Fonctions visibles sur Ω_H Fonctions invisibles sur Ω_H

x :points de la grille fine

● : points communs aux 2 grilles

Les fonctions propres de $\Delta(u)=f$: $\sin(k\pi x)$ k=1, ,7

Figure V-23 : Comportement des fonctions propres sur les deux grilles

(schéma inspiré de [MG 2])

Nous avons placé sur la colonne de gauche les "hautes fréquences" et sur celle de droite les "basses". Comme on peut le constater, chaque fonctions "haute fréquence" a sur la grille grossière (points cerclés en noir) les mêmes valeurs (au signe près) que la fonction "basse fréquence" située sur la même ligne. Il reste aussi une fonction qui sur la grille grossière est identiquement nulle. Le lecteur pourra aussi constater que si l'on avait pris un point sur quatre (au lieu d'un point sur deux), il ne serait plus resté qu'une seule fonction propre.

Une des tendances naturelles dans l'utilisation des méthodes multigrilles est de choisir une stratégie de réduction des grilles diminuant au maximum leurs tailles. On diminue ainsi le nombre de grilles et donc le nombre d'opérations de transfert très coûteuses en temps de calcul. Malheureusement, en contrepartie, le nombre des fonctions "basses fréquences" est très diminué. Cette étude a été généralisée en 2D par Trottenberg [MG 2] sur le même type de problème $\Delta(u(x,y)) = f(x,y)$ avec $(x,y) \in [0,1]^2$ par différences finies.

La conséquence importante que nous pouvons en tirer est que pour un changement de grille donné, le lissage doit être fait de façon suffisante pour détruire les "hautes fréquences". Si par malheur, ce n'est pas le cas, il ne faudrait pas compter sur la correction sur grille grossière pour le faire; sur cette grille, les "hautes fréquences" sont invisibles.

Un lissage et une correction mal coordonnés peuvent donc facilement faire diverger la méthode multigrilles. Cela nous incite une fois de plus à apporter le maximum de soin au lissage, et ce au moindre coût possible.

Une petite illustration de ce genre de problème. Le programme test étudie une poutre cantilever par éléments finis (figure V-13). Le solveur est une méthode 3-grilles utilisant un lissage par Gauss-Seidel relaxé (de paramètre de relaxation ω). Nous comparons avec un solveur par Gauss-Seidel relaxé. Notre facteur d'étude est le paramètre ω dans les deux cas.

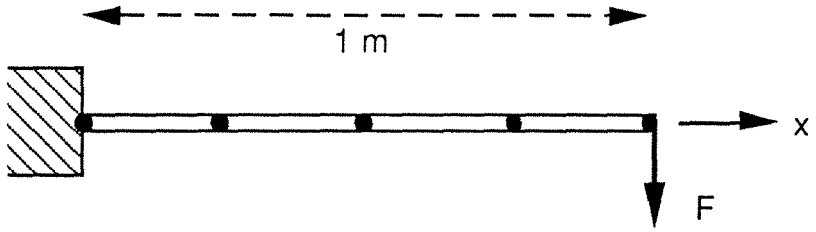


Figure V-13 : Maillage du problème (éléments de Bernouilli 2-nœuds)

On obtient les résultats suivants.

Solveur	Domaine de convergence pour ω	ω optimal	nombre d'itérations pour avoir $\|\text{erreur}\|_{sup} \leq 1\ \%$
méthode 3-grilles	] 0 , 1.1 [	0,7	7
Gauss Seidel	] 0 , 2 [	1,99	650

Tableau V-3 : Comparaison des domaines de validité de ω

Il ressort de nos essais que le paramètre ω joue un rôle différent dans chaque méthode. En particulier, le $\omega_{optimal}$ de la seconde méthode fait complètement diverger la première. L'explication en est assez simple : la meilleure efficacité en tant que solveur de traduit par un moindre effet en tant que lisseur; de fait, la non destruction des erreurs "hautes fréquences" fait diverger la méthode.

V-6 : Méthode Full Multigrilles

V-6-1 : présentation

C'est la meilleure façon d'utiliser les méthodes multigrilles. Comme toute méthode itérative, l'efficacité du processus multigrilles dépend de la valeur de départ prise par le vecteur inconnu. L'un des principaux intérêts du processus full-multigrilles est de construire lui même cette valeur de départ. En voici l'illustration :

Soit un processus (l+1)-grilles numérotés de 0 à l , Ω_0 étant le plus grossière, Ω_l étant la plus fine. A tous niveaux, nous supposerons que le problème à résoudre peut se mettre sous la forme matricielle habituelle :

$$[L_{hi}]\{u_i\} = \{f_{hi}\} \qquad (37)$$

Sur Ω_0, on peut aisément résoudre le problème vu sa taille, soit obtenir :

$$\{u_0\} = [L_{h0}]^{-1}\{f_{h0}\} \qquad (38)$$

on peut dès lors obtenir $\{\tilde{u}_1\} = p(\{u_0\})$, (p désignant une prolongation)
$\{\tilde{u}_1\}$ constitue une approximation assez grossière de la solution du problème sur Ω_1.
Pour améliorer cela, on applique quelques itérations d'une méthode 2-grilles (entre Ω_1 et Ω_0) en prenant $\{\tilde{u}_1\}$ valeur de départ ; ce qui donne une valeur $\{\hat{u}_1\}$ bien meilleure.

A partir de là, on calcule $\{\tilde{u}_2\} = p(\{\hat{u}_1\})$ et l'on recommence pour $\{\tilde{u}_2\}$ ce qui à été fait pour $\{\tilde{u}_1\}$ et ainsi de suite jusqu'au niveau (l-1).

Arrivé sur la grille W_l par $\{\tilde{u}_l\} = p(\{\hat{u}_{l-1}\})$, on dispose dès lors d'une bonne valeur de départ pour lancer le processus l-grilles.

On peut résumer la méthode Full-multigrilles par le schéma suivant :

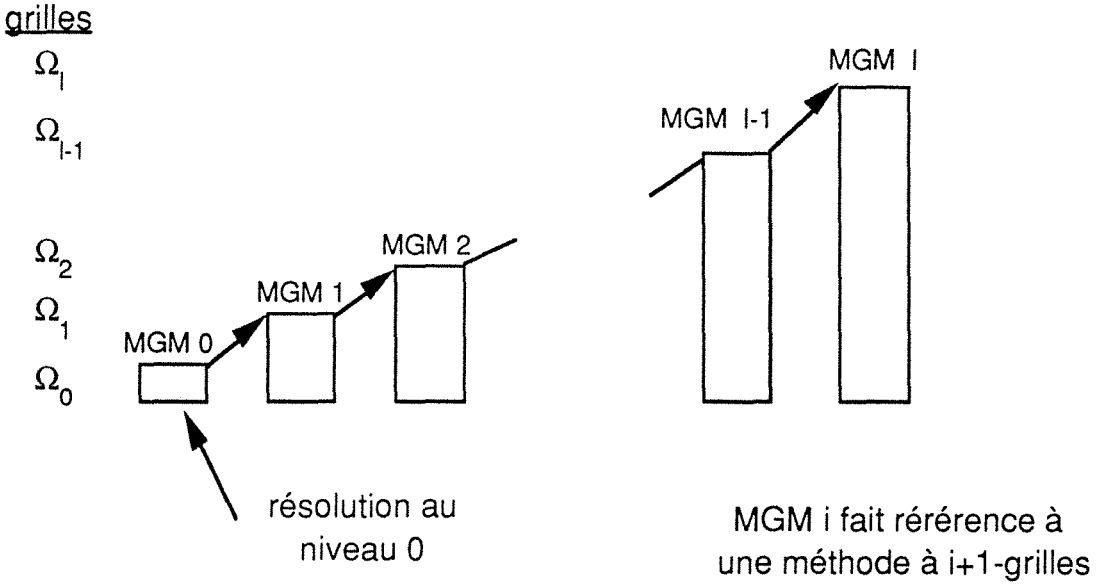


Figure V-14 : Méthode Full-multigrilles

L'algorithme de la méthode est alors :

a) _ Initialisation : $\{u_0\} = [L_{h0}]^{-1} \{f_{h0}\}$ et $\{\hat{u}_0\} = \{u_0\}$

b) _ Pour k:=1 Jusqu'à l Faire

_ Initialisation : $\{\tilde{u}_k\} = p(\{\hat{u}_{k-1}\})$

_ Processus itératif : on effectue IT itérations d'un cycle k-grilles avec $\{u_k\}$ comme valeur initiale

Pour j=1 Jusqu'à IT Faire MGM (k, $\{u_k\}$, $\{f_{hk}\}$)

Fin de la boucle

Tableau V-4 : cycle Full-multigrille

La valeur de IT peut être fixée ou bien contrôlée dynamiquement sur chaque grille.
La méthode Full-multigrilles présente les avantages suivants :

1) _ Le nombre de grilles l peut ne pas être fixé à l'avance; cela permet de raffiner dynamiquement le maillage.
2) _ On obtient de très bons résultats en non-linéaire où il est important d'avoir une bonne valeur de départ pour converger rapidement.

V-6-2 : Comparaison de diverse méthodes entre-elles

Trottenberg [MG 2] propose une étude contradictoire portant sur l'étude du problème classique de l'équation de poisson bidimensionnelle par diverses méthodes ; soit :

$$\begin{cases} \Delta(u(x,y)) = f(x,y) \quad \text{sur } \Omega = [0,1]^2 \\ \quad \text{avec } u(x,y) = g(x,y) \quad \text{sur } \delta\Omega \end{cases} \qquad (39)$$

La discrétisation du Laplacien se fait par un schéma de différences finies 5-points avec un pas de h=1/256 ; On se ramène alors à la résolution d'un système carré d'ordre 65025. On itère pour chaque méthode jusqu'à obtenir une erreur de l'ordre de l'erreur systématique $\varepsilon = h^2 \cong 10^{-4}$. On obtient les résultats suivants :

méthode	coût assymptotique	temps de calcul (IBM 360 158)
Gauss Seidel	$N^2.\log(N)$	≈ 20 heures
MICCG	$N^{5/4}.\log(N)$	≈ 6 minutes
SOR optimal	$N^{3/2}.\log(N)$	≈ 20 minutes
MG standart	N.log(N)	≈ 18 secondes
FULL MG	N	≈ 7 secondes

Tableau V-5 : Comparaison des résultats pour diverses méthodes

On constate un coût asymptotique de l'ordre de N pour la méthode Full-multigrilles, ce qui est optimal dans notre cas et prouve la grande puissance de la méthode.

V-7 : Problèmes non-linéaires

V-7-1 : Utilisation des multigrilles dans le cadre d'une méthode de Newton

Nous avons maintenant à résoudre un problème non-linéaire. En nous donnant une grille Ω_h et un schéma de discrétisation, nous obtenons, après introduction des conditions limites, un système non-linéaire à résoudre de la forme :

$$\mathbb{L}_h(\{u_h\}) = \{f_h\} \qquad (40)$$

Résoudre ce système par une méthode de linéarisation de Newton conduit au processus itératif suivant :

$$\{u_h^{j+1}\} = \{u_h^j\} + \{v_h^j\} \qquad (41)$$

où $\{v_h^j\}$ est solution de $[L_h^j].\{v_h^j\} = \{f_h\} - \mathbb{L}_h(\{u_h^j\})$ (41')

où :

- $\{u_h^j\}$ est l'itération numéro i
- $\{u_h\}$ est la solution exacte
- $\{v_h^i\} = \{u_h\} - \{u_h^i\}$ est l'erreur à l'itération i
- $[L_h^j] = \mathbb{L}'_h(\{u_h^j\})$ est l'opérateur linéaire tangent.

Durant chaque pas de cette méthode, un processus multigrilles linéaire peut être utilisé pour la résolution de l'équation linéaire (41').

A partir de là, deux écoles sont possibles :

_ La première consiste à adopter à chaque pas le nombre d'itérations multigrilles pour exploiter la vitesse de convergence de la méthode de Newton aussi loin que possible. Ainsi, pour avoir une convergence en norme quadratique, il faudra en gros doubler le nombre d'itérations multigrilles d'un pas à l'autre. L'inconvénient de cette méthode est qu'elle nécessite un contrôle à chaque pas.

_ La seconde, au contraire consiste à fixer le nombre de pas (1 à 3 par exemple). Cependant, cette variante n'assure qu'une convergence linéaire (celle de la méthode multigrilles) et de plus, le travail de linéarisation est plus important.

V-7-2 : Méthode FAS (Full approximation Scheme)

Les deux méthode que nous venons de voir ne sont que l'adaptation des multigrilles à la méthode de Newton. Il existe une variante plus proprement multigrilles : la méthode FAS.

Cette méthode reprend le fondement de la méthode multigrilles classiques, c'est à dire le couplage lissage-correction. Mais ici l'équation de correction ne peut plus être écrite aussi simplement à cause de la non-linéarité ::

si l'on note :

_ $\{u_h^j\}$ est l'itération numéro i

_ $\{u_h\}$ est la solution exacte

_ $\{v_h^i\} = \{u_h\} - \{u_h^i\}$ est l'erreur à l'itération i

_ Ω_h et Ω_H les grilles fines et grossières respectivement

Après lissage, on aura sur Ω_h :

$$\{f_h\} - \mathbb{L}_h\left(\{u_h^{\sim j}\}\right) = \mathbb{L}_h\left(\{u_h^{\sim j}\} + \{v_h^j\}\right) - \mathbb{L}_h\left(\{u_h^{\sim j}\}\right) = \{d_h^j\} \qquad (42)$$

Comme d'habitude, nous allons essayer de résoudre ce système sur une grille plus grossière Ω_H. On aura alors sur Ω_H

$$\mathbb{L}_H\left(\left\{\begin{matrix}\sim j\\ u_H\end{matrix}\right\} + \left\{\begin{matrix}j\\ v_H\end{matrix}\right\}\right) - \mathbb{L}_H\left(\left\{\begin{matrix}\sim j\\ u_H\end{matrix}\right\}\right) = \left\{\begin{matrix}j\\ d_H\end{matrix}\right\} \qquad (43)$$

$$\text{avec } \left\{\begin{matrix}j\\ d_H\end{matrix}\right\} = r\left(\left\{\begin{matrix}j\\ d_h\end{matrix}\right\}\right) \text{ et } \left\{\begin{matrix}\sim j\\ u_H\end{matrix}\right\} = r'\left(\left\{\begin{matrix}\sim j\\ u_h\end{matrix}\right\}\right)$$

L'équation de correction de la méthode FAS s'écrit donc sous la forme suivante : trouver $\left\{\begin{matrix}j\\ w_H\end{matrix}\right\} = \left\{\begin{matrix}\sim j\\ u_H\end{matrix}\right\} + \left\{\begin{matrix}j\\ v_H\end{matrix}\right\}$ tel que :

$$\mathbb{L}_H\left(\left\{\begin{matrix}j\\ w_H\end{matrix}\right\}\right) = \left\{\begin{matrix}j\\ d_H\end{matrix}\right\} + \mathbb{L}_H\left(\left\{\begin{matrix}\sim j\\ u_H\end{matrix}\right\}\right) \qquad (44)$$

L'aspect algorithmique d'un processus FAS est alors le suivant :

I . Pré-lissage

$$\overset{\sim j}{u_h} = S_h^{\nu 1}\left(\left\{\begin{matrix}j\\ u_h\end{matrix}\right\}, \{f_h\}, \mathbb{L}_h\right)$$

II . Correction sur grille grossière

a) $\left\{\begin{matrix}j\\ d_h\end{matrix}\right\} = \{f_h\} - \mathbb{L}_h\left(\left\{\begin{matrix}\sim j\\ u_h\end{matrix}\right\}\right)$ calcul du défaut

b) $\left\{\begin{matrix}j\\ d_H\end{matrix}\right\} = r\left(\left\{\begin{matrix}j\\ d_h\end{matrix}\right\}\right)$ restriction du défaut

c) $\left\{\begin{matrix}\sim j\\ u_H\end{matrix}\right\} = r'\left(\left\{\begin{matrix}\sim j\\ u_h\end{matrix}\right\}\right)$ restriction de u

d) calcul de $\left\{\begin{matrix}j\\ w_h\end{matrix}\right\}$ solution de :

$$\mathbb{L}_H\left(\left[\begin{matrix}j\\ w_H\end{matrix}\right]\right) = \left\{\begin{matrix}j\\ d_H\end{matrix}\right\} + \mathbb{L}_H\left(\left[\begin{matrix}\sim j\\ u_H\end{matrix}\right]\right)$$

par $\gamma \leq 1$ pas de la méthode FAS en partant de la grille Ω_H avec $\left\{\begin{matrix}\sim j\\ u_H\end{matrix}\right\}$ comme première approximation .

e) $\left\{\begin{matrix}\sim j\\ v_h\end{matrix}\right\} = p\left(\left\{\begin{matrix}j\\ v_H\end{matrix}\right\}\right)$ prolongation de la correction

f) calcul de $\left\{ \tilde{v}_h^j + \tilde{u}_h^j \right\}$ sur Ω_h

III . Post-lissage

$$u_h^{j+1} = S_h^{v2}\left(\left\{ \tilde{u}_h^j + \tilde{v}_h^j \right\}, \{f_h\}, \mathbb{L}_h \right)$$

On notera qu'aucun travail de linéarisation n'est nécessaire sauf peut-être au niveau le plus grossier. On remarquera aussi que les restrictions du défaut et de u ne sont pas forcément les mêmes, ce qui permet des accommodements.

Les composantes de la méthode restent à peu près le mêmes sauf que les processus de lissage sont non-linéaires (Gauss ou Jacobi non-linéaires par exemple).

V-8 : Conclusion

Ce chapitre a eu pour but de familiariser le lecteur avec le comportement (parfois curieux) des méthodes multigrilles. On l'aura remarqué, certains faits sont plus des constatations que des démonstrations; l'aspect théorique du comportement de la méthode n'ayant pu être étudié que sur quelques cas simples. Il n'en reste pas moins que ces constatations fournissent de précieuses informations sur le comportement de la méthode et peuvent aider à améliorer son efficacité.

Cependant, comme on aura pu le constater, la méthode multigrilles et ses variantes peuvent s'avérer très efficaces. Cependant, cette efficacité n'est acquise que par un ajustement judicieux et précis des divers éléments composant la méthode.

L'écriture d'un code général d'éléments finis utilisant les méthodes multigrilles est donc assez difficile à réaliser, le nombre des paramètres intervenant (choix des grilles, des transferts entre grilles, des lisseurs et solveur,...) est élevé et de plus, ces paramètres ne sont pas indépendants. Toutes ces possibilités alourdissent énormément la programmation et nécessitent un utilisateur familiarisé avec les méthodes multigrilles. Il faut donc faire des choix.

Dans cette optique, la méthode Full-multigrilles semble apporter un net progrès de par son aspect dynamique, son automatisme et son efficacité. Utilisée judicieusement cette méthode doit permettre d'adapter le cycle de résolution au problème traité, nous ne donnerons un exemple au chapitre VIII.

Les méthode multigrilles n'en restent pas moins, encore maintenant, un sujet d'études intéressant et prometteur. La plupart des études menées l'on été dans le cadre de problèmes elliptiques. Les problèmes hyperboliques posent encore aujourd'hui beaucoup de problèmes théoriques et numériques.

Chapitre VI

Structure de données multigrilles

VI-1 : Introduction

Les problèmes posés par la construction d'un logiciel d'éléments finis sur micro-ordinateur sont aujourd'hui bien connus. Ils concernent essentiellement la gestion de l'espace mémoire, l'organisation des données et le choix de certaines méthodes. Leurs solutions sont elles aussi bien connues [EL 1], [EL 2], [EL 8], [EL 10]. Cependant, l'introduction des méthodes multigrilles pose de nouveaux problèmes dont les solutions sont parfois incompatibles avec les solutions précédentes. Nous présentons donc dans ce paragraphe l'ensemble des solutions que nous avons choisies et les raisons qui ont présidé à notre choix. Ces solutions n'ont nullement la prétention d'être définitives ou complètes, elles correspondent seulement à ce qui nous semblé être le plus judicieux compte-tenu des impératifs imposés.

La structure de données du programme est assujettie de façon permanente à deux problèmes duaux :ou bien l'on stocke (et l'on perd de la place mémoire), ou bien l'on retrouve (et l'on perd du temps de calcul). Nous avons selon le cas privilégié l'une ou l'autre des solutions selon l'optique que nous voulions donner au problème.

VI-2 : Choix de la méthode multigrilles

Il y aura là peu de suspense, surtout si l'on a lu le chapitre précédent, nous avons choisi la méthode Full-multigrilles pour ses performances et sa faculté d'adaptation au problème. Il va cependant falloir mettre au point une structure de données permettant de gérer sans trop de problèmes plusieurs grilles.

Le maillage de la grille la plus grossière est fourni par l'utilisateur. Les maillages des grilles ultérieures sont générées par divisions successives. Les opérateurs de prolongations pour passer d'une grille à une autre dans le processus Full-multigrilles sont les mêmes que ceux que nous utiliserons dans le processus multigrilles classique en lui-même.

A chaque niveau, la solution obtenue par prolongation sera améliorée par quelques itérations d'une méthode multigrilles en W-cycle utilisant l'ensemble des grilles déjà définies.

VI-3 : Stockage des points et des grilles

Les points sont stockés en mémoire centrale, par leurs coordonnées, dans leur ordre d'apparition au cours des créations de grilles. Cet ordre ne sera plus modifié par la suite. Il est à noter tout de suite qu'une telle numérotation est très désavantageuse au niveau de la largeur de bande, mais nous verrons plus loin comment nous avons résolu le problème.

Lorsqu'une nouvelle grille est construite, nous ne créons effectivement que les nouveaux points apparaissant. nous obtenons une structure similaire à celle ci-dessous (figure VI-1).

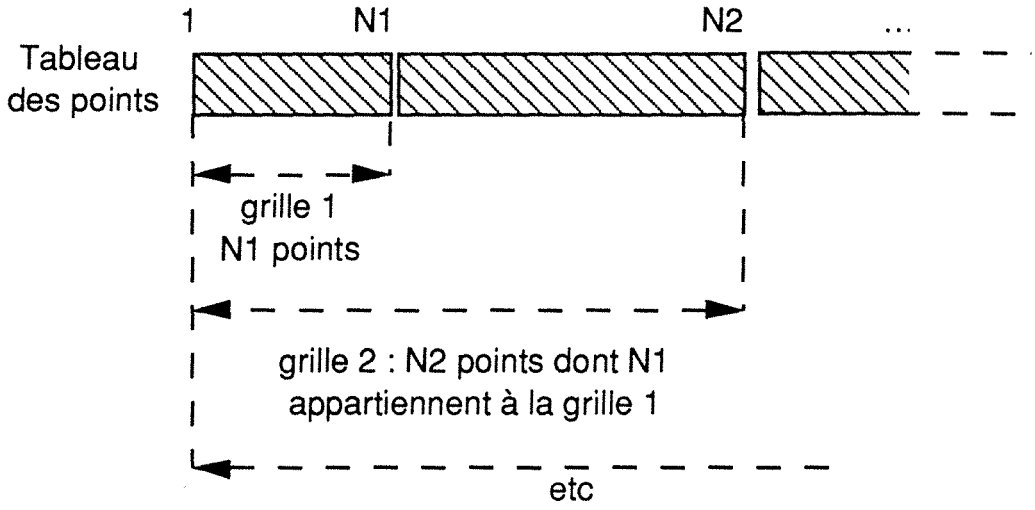


Figure VI-1 : stockage des points

Sur l'exemple de la figure VI-1, les points de numéros compris entre N_1 et N_2 appartiennent à la grille 2 mais pas à la grille 1. De ce fait, il est aisé à tous niveaux de savoir à quelles grilles appartient un point.

Les indicateurs N_1, N_2 , ...etc sont stockés dans un seul tableau en mémoire centrale.

Les éléments constituant les maillages de chaque grille sont stockés successivement de grille en grille sur fichier disque à accès direct. Chaque élément est défini par la donnée de ses nœuds parcouru dans le sens trigonométrique. A ceci s'ajoutent plusieurs champs utilisés par le solveur (épaisseur d'éléments,type de matériaux, etc). Nous obtenons finalement la structure suivante (figure VI-2) :

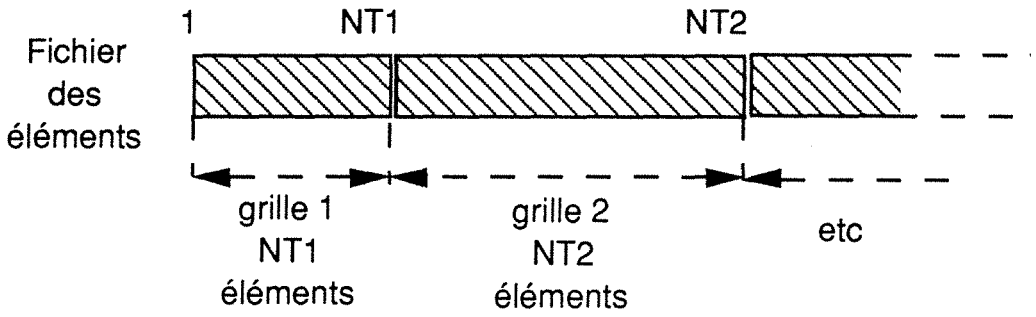


Figure VI-2 : Stockage des différentes grilles

La encore, la distinction des différentes grilles est aisée a effectuer.

Les arêtes et les côtés frontières de chaque grille sont stockées sur fichier disque à accès direct. Les définitions géométriques des arêtes frontières étant communes à toutes les grilles, celles-ci sont conservées une fois pour toutes. Les côtés frontières, eux varient au cours des grilles ; Ils sont donc crées dynamiquement et conservés selon une structure analogue à celle des éléments.

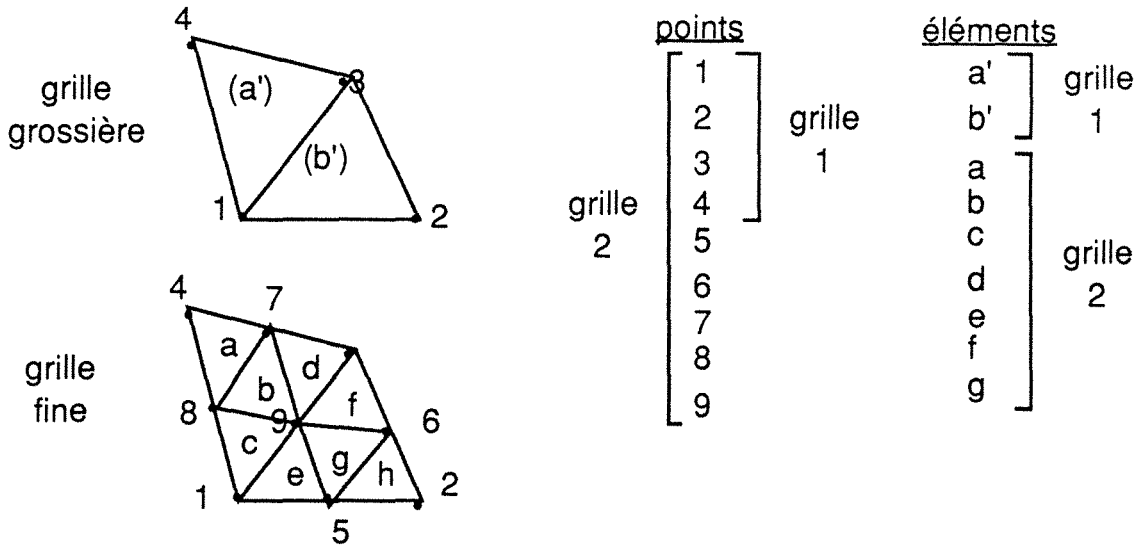


Figure VI-3 : exemple de stockage de grilles

VI-4 : Stockage des matrices de rigidité

C'est l'un des points-clés d'un logiciel d'éléments finis sur micro-ordinateur.

Les matrices issues des formulations par éléments finis sont très creuses. Il convient donc d'avoir une structure de données permettant le compactage des informations tout en permettant une relecture aisée à tous moments. La procédure la plus courante et la procédure "Ligne de ciel" [EL 2]. cependant, pour un même problème, le volume des informations stockées dépend fortement de la numérotation des points, cela nécessite donc un pré-processeur optimisant la largeur de bande. Sans cette précaution, la taille de la matrice devient énorme, surtout pour les grands systèmes, et de plus un très grand nombre des valeurs sont nulles.

Pour classique que soit cette structure, elle nous apparait ici tout à fait inadaptée pour les raisons suivantes :

1) _ Telles que sont construites nos grilles, la largeur de bande est quasiment maximale, on ne peut donc pas se passer d'optimisation.
2) _ Vue la multiplicité des grilles; il est nécessaire d'optimiser la largeur de bande sur chaque grille. Cela nécessite donc un calcul d'optimisation et le stockage permanent de la renumérotation pour chaque grille. Donc du temps de calcul et de la place mémoire perdue pour chaque grille.

Nous avons donc préféré la méthode Morse [EL 8] [EL 10] qui offre un

compactage optimal de la matrice et ce indépendamment de la numérotation des nœuds. Rappelons-en le principe.

Par construction, les matrices issues d'une formulation par éléments finis ont les propriétés suivantes :

_ La topologie de la matrice (ie la position des termes non-nuls) est prévisible. Pour cela, $a_{ij} \neq 0$ si i et j sont des degrés de liberté correspondant à des nœuds appartenant à un même élément.

_ La topologie de la matrice est symétrique. Par contre, la matrice en elle-même ne l'est pas nécessairement (bien que ce soit très souvent le cas). De plus, les matrices sont souvent définies positives. Dans la suite, nous supposerons que nous n'avons plus affaire qu'à des matrices symétriques définies positives.

Le processus de mise en place du stockage Morse se déroule en deux étapes :

1) _ Mise en place de la topologie

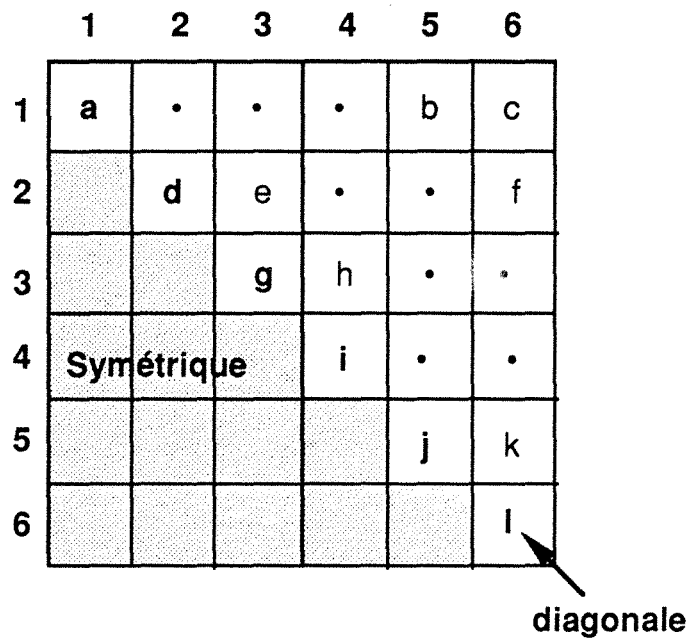
* A partir de la connaissance des connections de chaque élément, on établit, pour chaque point, la liste de ses voisins classées dans l'ordre croissant. Comme notre matrice est symétrique, on ne s'intéressera qu'aux voisins de numéro supérieur ou égal à celui du point (chaque nœud est lié à lui-même).
* L'ensemble est stocké dans deux tableaux unicolonnes (nommés IA et JA). Le premier contient, mis bout à bout, l'ensemble des voisins de chaque point. Le second contient un ensemble d'adresses pointant sur le tableau précédent et permettant d'extraire à tous moment l'ensemble des voisins d'un nœud i donné. Cet ensemble constitue l'ensemble des emplacements des termes non-nuls de la ligne i (parties supérieure).

2) _ Assemblage proprement dit

* La matrice de rigidité élémentaire de chaque élément est évalué
* A partir des deux tableaux construits en 1), on retrouve l'adresse réelle de chaque terme élémentaire dans la matrice globale. La matrice globale est stockée sous la forme d'un tableau unicolonne AK ayant la même taille que IA. Ce dernier tableau est géré par les deux tableaux de pointeurs IA

et JA. JA permet de séparer entre elles chaque ligne et IA permet, pour chaque ligne, de "décompacter" les termes et de retrouver leur position réelle. On ajoute donc dans la matrice globale à la place correcte, la contribution élémentaire de chaque élément.

Pour illustrer le processus, prenons l'exemple d'une matrice 6x6 (figures VI-4 a à VI-4 c).



Légende
a,...,l : terme non nul
• : terme nul

Figure VI-4 a : Matrice de départ

Le fichier des voisins est alors :

point	voisins
1	1 5 6
2	2 3 6
3	3 4
4	4
5	5 6
6	6

Figure VI-4 b : Le fichier des voisins

On obtient les 3 tableaux suivants :

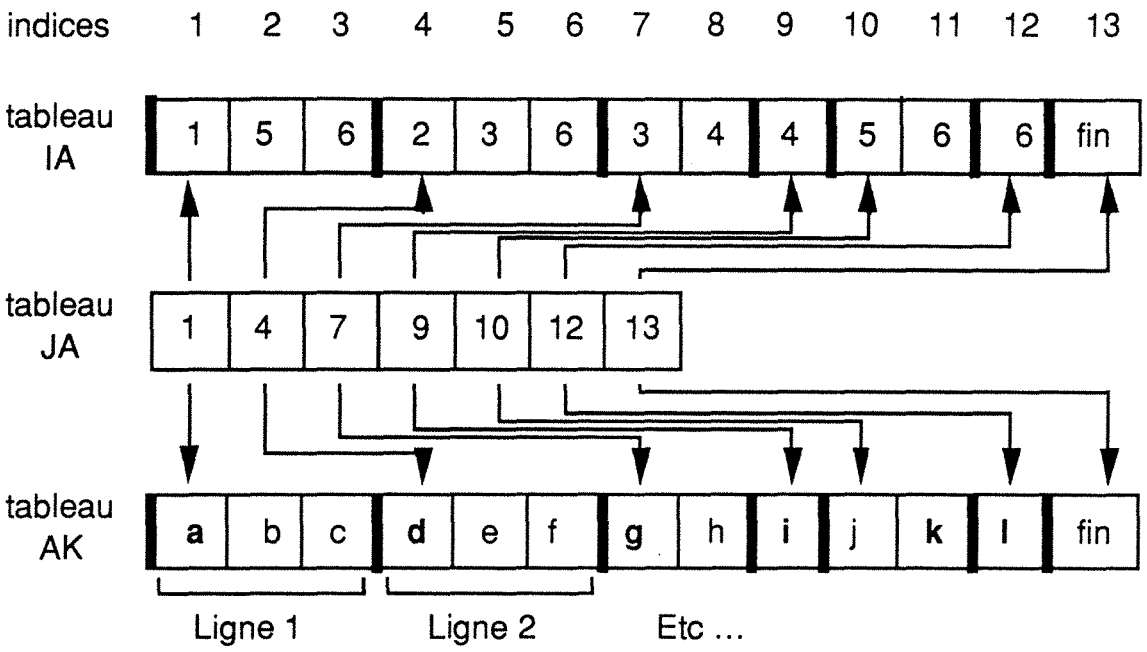


Figure VI-4 c : Le résultat final du stockage

* On notera le rôle central du tableau JA. Celui ci permet de repérer les débuts et fins de chaque ligne, ainsi que la position du terme diagonal.
* Il est possible de stocker les termes diagonaux à part, ce qui fait gagner un peu de place mais qui nécessite un tableau de plus.
* Dans le cas où la matrice n'est plus symétrique, la remarque précédente est utilisée. La matrice en elle-même n'est plus représentée par un seul tableau mais par trois (diagonale, partie triangulaire inférieure, partie triangulaire supérieure) , cependant ces trois tableaux sont gérés par la même topologie (IA et JA).

Présentons pour finir les deux arguments-massues de ce type de stockage :

1) _ Le nombre de termes stocké est indépendant de la numérotation des nœuds. Point n'est besoin d'optimisation de la numérotation, ce qui convient bien à notre structure de stockage des grilles.
2) _ Lorsque la taille de la matrice tend vers l'infini, il est prouvé que ce stockage tend vers un nombre de termes mémorisés optimal.

VI-5 : Opérateurs de lissage

Nous avons choisi un lissage par Gauss-Seidel relaxé. Cette méthode est efficace et aisée à mettre en œuvre de façon informatique (rien à stocker hormis la matrice de rigidité).

si l'on écrit la matrice de rigidité $[K] = [D] + [E] + [E]^t$

où [D} est la matrice formée par la diagonale de [K]

et [E] est la partie triangulaire inférieure de [K]

On peut alors écrire l'itération (i+1) du lissage en fonction de l'itération i par l'équation :

$$\{u\}^{i+1} = \left([D] + \omega[E] \right)^{-1} . \left(\left((1 - \omega).[D] - \omega.[E]^t \right).\{u\}^i + \omega.\{f\} \right) \quad (1)$$

On notera la simplicité du calcul à effectuer, d'autant plus qu'avec le type de stockage utilisé, les produits matrice-vecteur sont effectués rapidement et ne concernent que les termes non-nuls.

Le choix du paramètre de relaxation ω a été de ω=0,5. Cette valeur , essentiellement issue de la pratique, a donné de bons résultats pour le lissage.

VI-6 : Mise en place des divisions de grilles

Le problème est de mettre en place une structure permettant le découpage d'éléments de types différents et découpés de façons différentes. Les divisions d'éléments, comme nous le verrons plus loin, peuvent être faites de façon homogène (tous les éléments découpés de la même façon) ou non-homogène.

Une fois choisi le type de division, l'élément est découpé via l'élément de référence. Ceci permet d'avoir, pour chaque type de division, une référence unique donnant le nombre et la place des points crées ainsi que la topologie des nouveaux éléments crées.

Cela permet à tous moment :

- de savoir quels points sont apparus et quels sont leurs coordonnées (de référence ou réelles).
- de savoir quels sont les éléments nouveaux qui sont apparus.

Ainsi, le passage géométrique d'une grille à une autre est aisé et cela facilite prolongations et restrictions.

Voici quelques exemples de découpages (figure VI-5) :

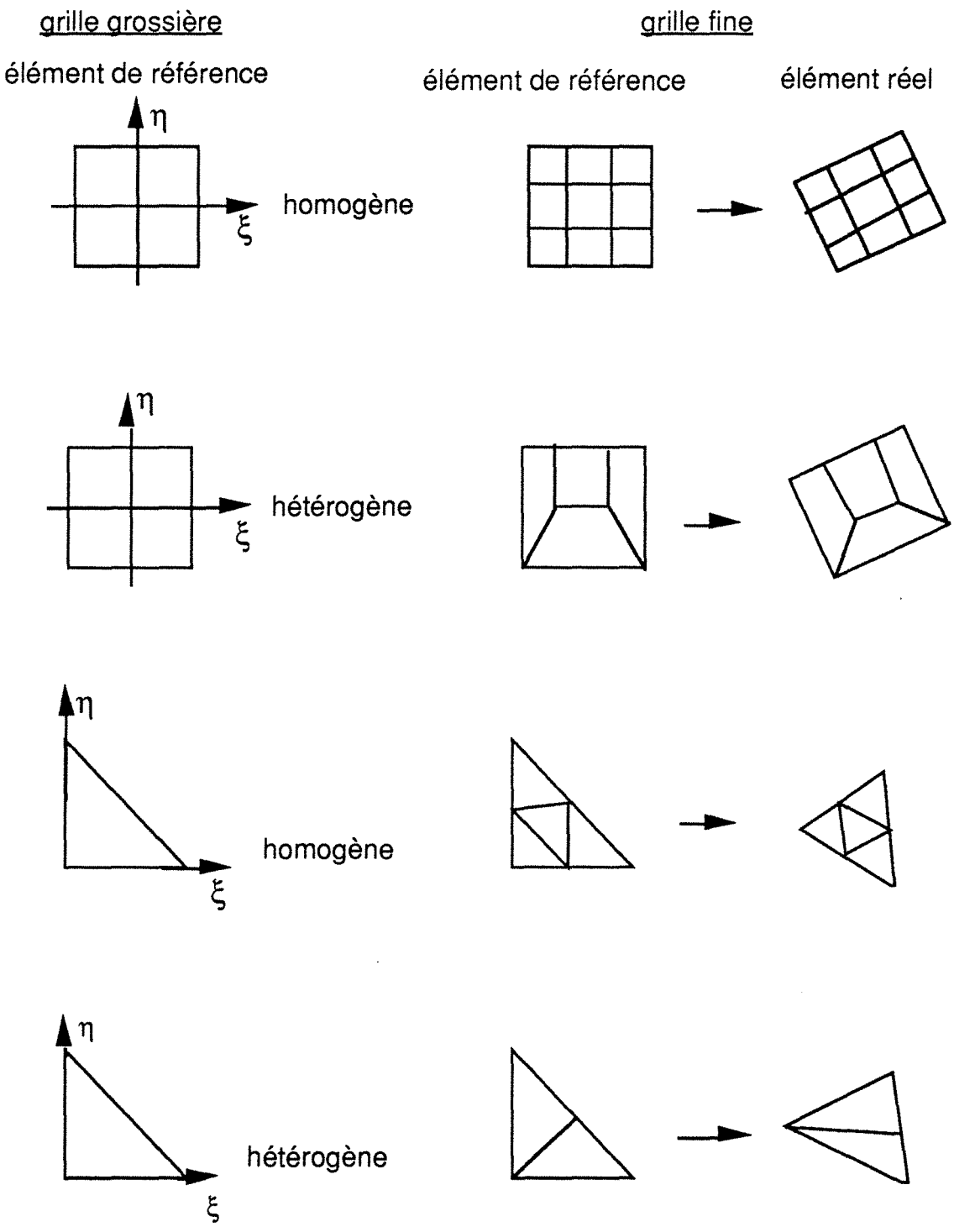


Figure VI-5 : Exemples de divisions

VI-7 : Prolongations et restrictions

Nous supposerons pour la suite de ce paragraphe, que nous souhaitons soit prolonger soit restreindre un vecteur entre une grille fine et une grille grossière.

Nous allons utiliser dans les deux cas, les propriétés d'approximation nodale intervenant dans la formulation par éléments finis.

* Prolongation

Deux types de points se présentent :

_ ceux qui sont communs aux deux grille

_ ceux qui n'appartiennent qu'à la grille fine

Les points de la seconde catégorie apparaissent lors de la division des éléments de la grille grossière et sont donc connus (comme nous l'avons vu en VI-6) par leurs coordonnées réelles et de référence.

Nous allons donc travailler élément par élément en effectuant sur chaque élément :

$$\text{pour un élément} \quad \begin{cases} u_i^{GF} = u_i^{GG} \qquad i=1,..,NP \\ u_p^{GF} = \sum_{i=1}^{NP} N_i(\xi_p, \eta_p) \cdot u_i^{GG} \qquad p=1,..,NQ \end{cases} \tag{2}$$

_ les indices $\cdot^{GF}$ et $\cdot^{GG}$ indiquent les valeurs du vecteur sur les grilles fines et grossières respectivement.

_ NP est le nombre de nœuds de l'élément de la grille grossière.

_ NQ est le nombre de nœuds apparaissant lors de la division de l'éléments de la grille grossière.

_ N_i sont les NP fonctions de forme de l'élément de la grille grossière (i=1,NP)

_ (ξ_p, η_p) sont les coordonnées de référence du point p (p=1,NQ)

* Restrictions

Nous choisissons une restriction telle que $r = p^t$

Nous obtenons donc :

$$u_i^{GG} = u_i^{GF} + \sum_{p=1}^{NQ} N_i(\xi_p, \eta_p) . u_p^{GF} \qquad (3)$$

(mêmes notations que précédemment)

Nous fournissons ci-dessous un exemple dans le cas d'un élément 4-nœuds divisé en 4 :

ici : NP = 4 (points 1 à 4 marqués par •)

NQ = 5 (points 1' à 5' marqués par x)

→ sens du transfert des informations

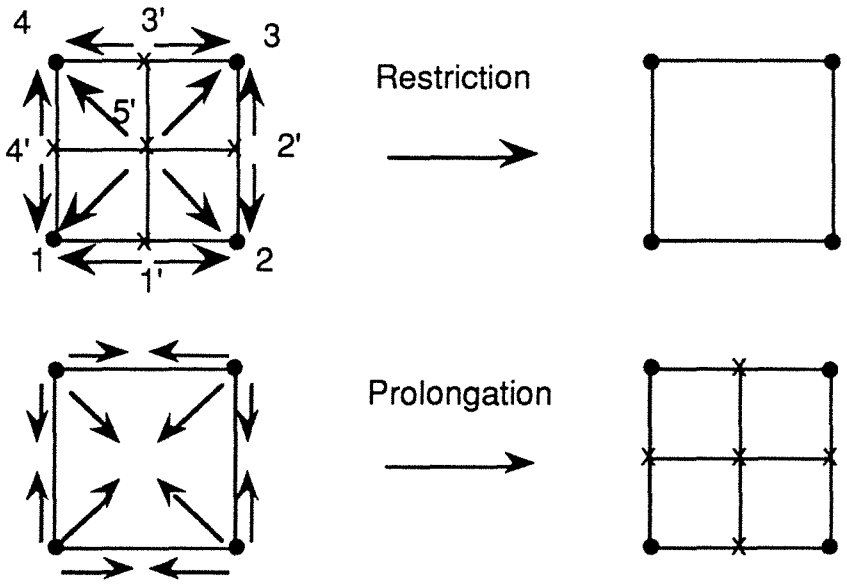


Figure VI-6 : exemple de prolongation et de restriction

Dans ce cas précis, on obtient pour la prolongation :

$$u_{1'}^{GF} = \frac{1}{2}\left(u_1^{GG} + u_2^{GG} \right)$$

$$u_{2'}^{GF} = \frac{1}{2}\left(u_1^{GG} + u_4^{GG} \right)$$

$$u_{4'}^{GF} = \frac{1}{2}\left(u_2^{GG} + u_3^{GG} \right)$$

$$u_{5'}^{GF} = \frac{1}{2}\left(u_3^{GG} + u_4^{GG} \right)$$

$$u_{3'}^{GF} = \frac{1}{4}\left(u_1^{GG} + u_2^{GG} + u_3^{GG} + u_4^{GG} \right)$$

Les points appartenant à plusieurs éléments à la fois ne sont prolongés et corrigés qu'une seule fois.

Ces prolongation sont données dans le cadre d'éléments classiques utilisant une interpolation de Lagrange. Les éléments de type Hermitte sont prolongés d'une façon différente utilisant les propriétés de l'interpolation. Ceci sera vu plus en détail dans le prochain chapitre (chapitre VII).

VI-8 : Vecteur inconnu U et vecteur second membre F

Ces deux vecteurs sont stockés en mémoire centrale sous la forme d'un tableau unicolonne. Pour satisfaire à la définition dynamique des grilles, les informations relatives à chaque grille sont mise bout à bout (de façon identique au stockage des éléments vu au paragraphe VI-3). Un jeu d'adresses commun au deux vecteurs permet de s'y retrouver.

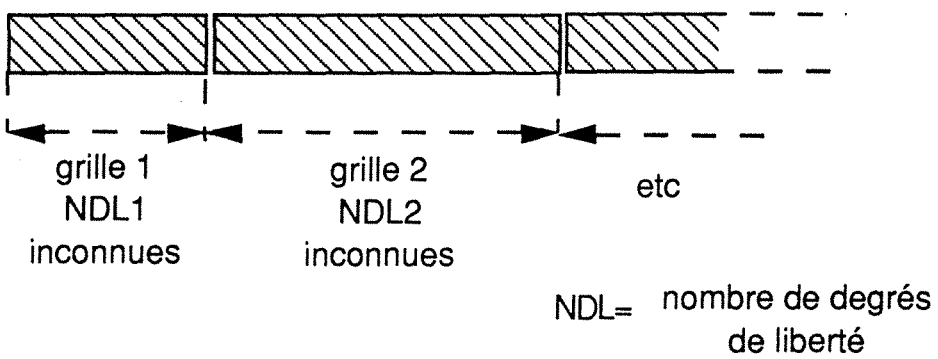
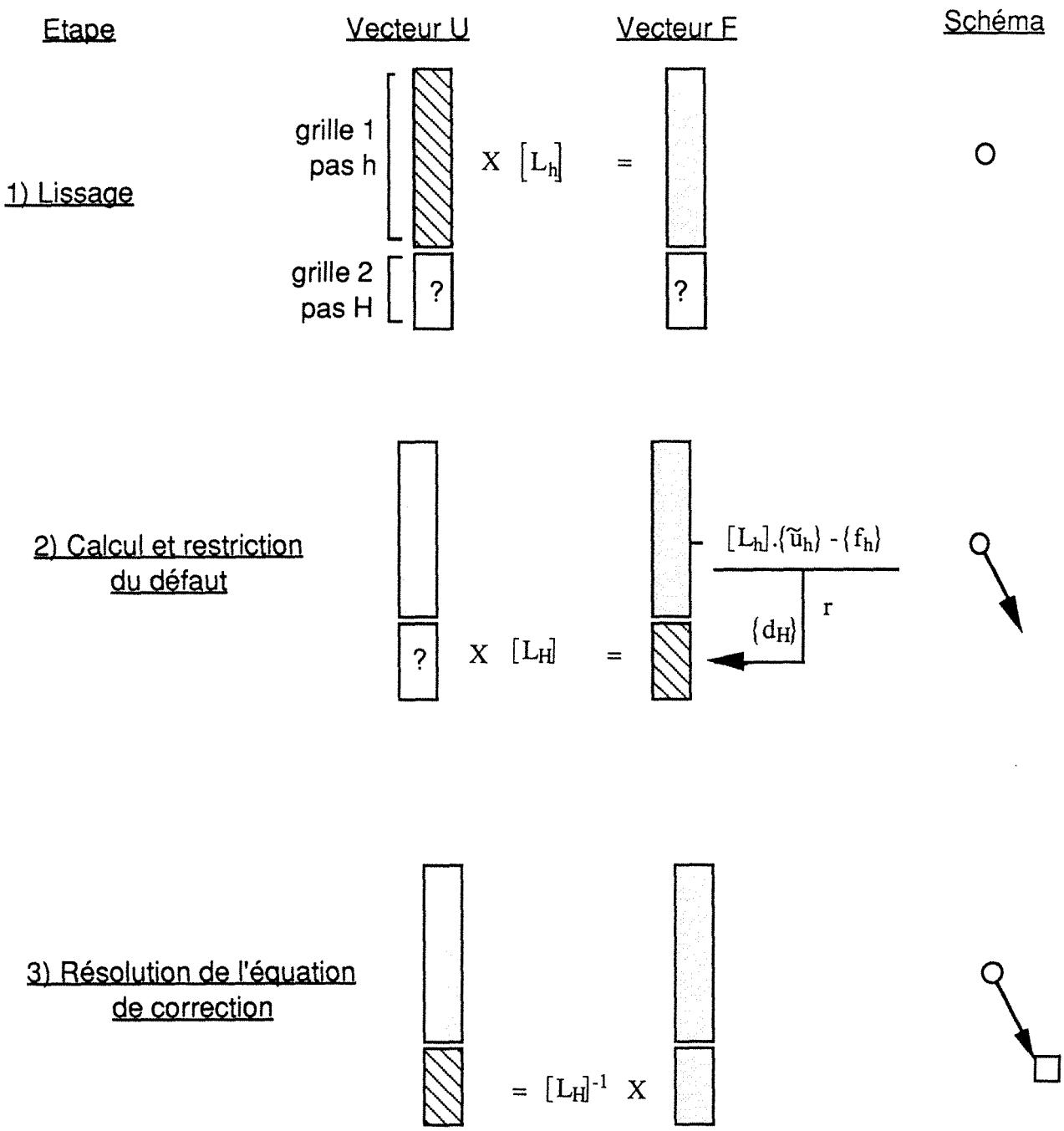


Figure VI-7 : stockage de U et F

Pour un cycle 2-grilles les vecteurs U et F évaluent de la façon suivante :



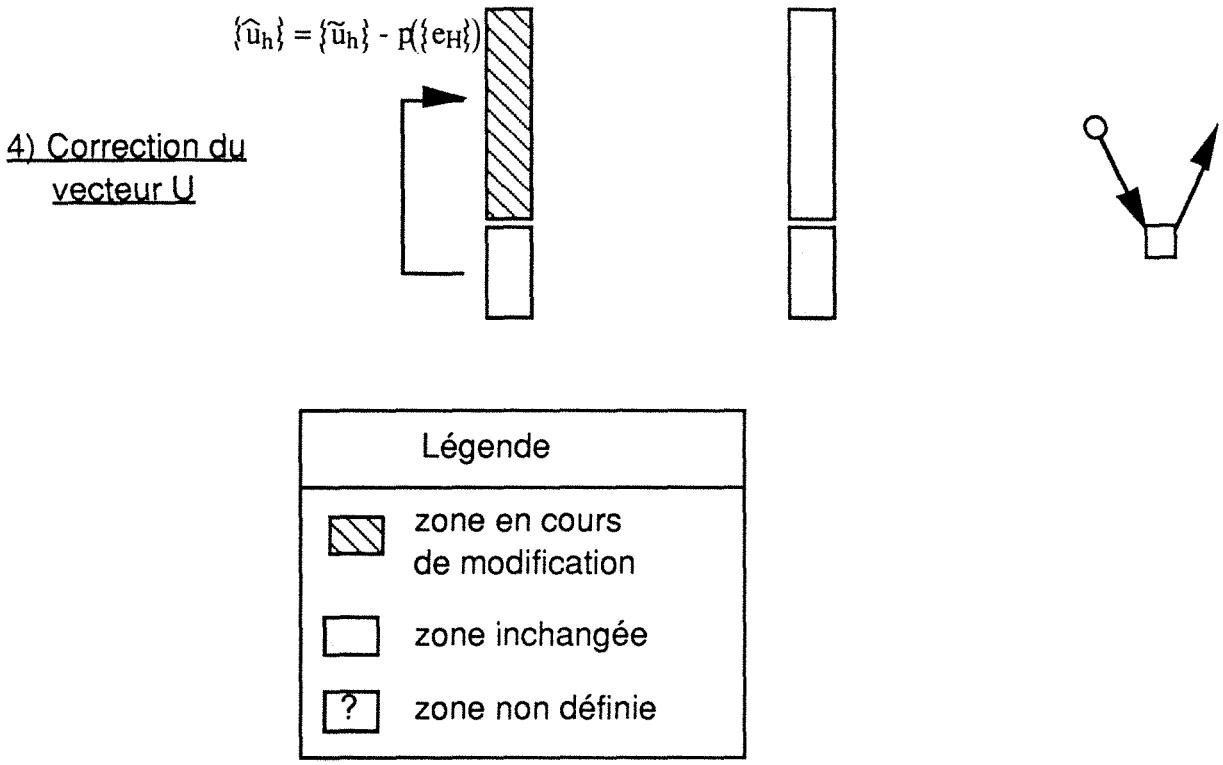


Figure VI-8 : Evolution des vecteurs U et F dans un cycle 2-grilles

VI-9 : Matrice de rigidité sur la grille grossière

Pour des raisons de simplicité et de stockage, nous avons décidé de prendre pour $[L_H]$ la matrice de rigidité construite à partir sur le maillage de la grille grossière par la formulation du problème par éléments finis.

Les matrices résident de façon permanente sur fichier disque paginé à accès direct. Lors du parcours de grilles, la matrice de la grille en cours est rappelée en mémoire centrale si cela est nécessaire (pour les lissage ou la résolution).

VI-10 : Résolution de l'équation de correction au niveau le plus grossier

Encore un point clé du logiciel : cette résolution doit se faire à moindre coût (en temps et place mémoire). Après avoir consulté les écritures saintes, il est apparu que les méthodes de gradient conjugués étaient les plus efficaces.

Nous avons effectué des essais contradictoires sur deux variantes semblant les plus efficaces

- _ Gradient conjugué avec préconditionnement par factorisation de Choleski incomplète (ICCG).
- _ Gradient conjugué avec préconditionnement par surrelaxation symétrique successive (SSORCG).

a) rappels sur les gradients conjugués

On souhaite résoudre le système $[K]\{u\} = \{f\}$ dans $\Re^n$. La matrice [K] est issue d'une formulation par éléments finis, et nous supposerons, comme c'est souvent le cas, qu'elle est symétrique définie positive. Ceci impliquera que $[K]^{-1}$ le sera aussi.

Soit [H] une autre matrice symétrique définie positive. Nous posons $\{r\} = \{f\} - [K].\{u\}$ (vecteur résidu).

La fonctionnelle $E : \{r\} \rightarrow < \{r\} , [K]^{-1}.\{r\} >$ est strictement convexe et admet un minimum pour $\{r\} = \{0\}$. Le but de la méthode est de minimiser itérativement E en construisant une base orthogonale ($\{p_0\},\ldots,\{p_{n-1}\}$) de $\Re^n$, orthogonale suivant un produit scalaire particulier, en minimisant E successivement suivant chaque direction. Le vecteur $\{u\}$ obtenu vérifiera $\{r\}=\{0\}$ soit $[K].\{u\} = \{f\}$.

En voici le processus

1) Initialisation

- $\{u_0\}$ quelconque
- $\{r_0\} = \{f\} - [K].\{u_0\}$
- $\{g_0\} = \{r_0\}$
- $\{p_0\} = [K].\{g_0\}$

2) Processus itératif

Pour k=1 à n-1 Faire

2-a) minimisation

- $\alpha_k = \langle\langle \{g_k\} , \{p_k\} \rangle\rangle / \langle [K] . \{p_k\} , \{p_k\} \rangle$
- $\{u_{k+1}\} = \{u_k\} + \alpha_k . \{p_k\}$
- $\{g_{k+1}\} = \{g_k\} - \alpha_k . [K] . \{p_k\}$

2-b) orthogonalisation

- $\beta_{k+1} = - \langle [H] . \{g_{k+1}\} , [K] . \{p_k\} \rangle / \langle [K] . \{p_k\} , \{p_k\} \rangle$
- $\{p_{k+1}\} = [H] . \{g_{k+1}\} - \beta_{k+1} . \{p_k\}$

Fin des itérations

Tableau VI-1 : Algorithme de Gradient conjugué

b) propriétés des gradients conjugués

On démontre les résultats suivants :

- 1) pour $o \leq i \leq j$ $\left\langle \{g_i\} , \{p_j\} \right\rangle = 0$
- 2) $\forall\, i \neq j$ $\left\langle [K].\{p_i\} , \{p_j\} \right\rangle = 0$ (conjugaison)
- 3) $\forall\, i \neq j$ $\left\langle \{g_i\} , [H].\{g_j\} \right\rangle = 0$
- 4) $\{u_{k+1}\}$ minimise E sur l'espace affine passant par $\{u_0\}$ et dirigé par l'espace vectoriel ($\{p_0\},\ldots,\{p_k\}$)
- 5) l'algorithme converge en au plus n itérations. Ceci permet de transformer un processus itératif en processus "semi-direct" puisque le nombre d'itérations se trouve borné.

c) Gradient conjugué simple

C'est le cas le plus simple. Il consiste à prendre [H] = [I]. Cependant, avec un tel choix, la rapidité de convergence dépend du conditionnement de [K]. On est donc amené à "pré-conditionner" la matrice [K] pour améliorer les performances de la méthode.

d) Gradient conjugué préconditionné

Le conditionnement intervient par l'équation

$$\{p_{k+1}\} = [H].\{g_{k+1}\} + \beta_{k+1}.\{p_{k+1}\}$$

Les critères menant au choix de [H] sont essentiellement basés sur des préoccupations informatiques et numériques :

_ Le calcul de l'inversion de [H] ne doit pas être trop coûteux

_ le stockage de [H] (qui à la même taille que [K]) ne doit pas être trop lourd.

Parmi tous les choix possibles, les plus courants sont :

- $[H]^{-1} = ([L].[L]^t)$ où [L] est la matrice issue de la factorisation de Choleski incomplète (factorisation sans tenir compte des termes du remplissage). Cette matrice [L] se calcule de la façon suivante :

Pour i=1 à n Faire

  Pour j=1 à i-1 Faire

    si $k_{ij} = 0$ alors $l_{ij} = 0$

    sinon $l_{ij} = \left(k_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{kj} \right) / l_{jj}$

  Fin de la boucle sur j

  $l_{ii} = \left(k_{ii} - \sum_{k=1}^{j-1} l_{ik}^2 \right)^{\frac{1}{2}}$

Fin de la boucle sur i

Avec un tel choix, la topologie de [L] (tableaux IA et JA vus en VI-4) est la même que celle de [K]. Cependant, la matrice [L] elle-même doit être stockée. Par ailleurs, si ce préconditionnement est des plus efficaces, il est malheureusement limité. En effet, les relations de factorisation impliquent certaines relations sur les k_{ij} (notamment $\left(k_{ii} - \sum_{k=1}^{j-1} l_{ik}^2 \right) > 0$). Ces relations ne sont pas toujours vérifiées , en particulier pour les éléments de degrés 2.

• Autre possibilité : le conditionnement SSOR. Il consiste à prendre :

$$[H]^{-1} = \frac{1}{2-\omega}\left(\frac{1}{\omega}[D]+[E]\right).\left(\frac{1}{\omega}[D]\right)^{-1}.\left(\frac{1}{\omega}[D]+[E]^{t}\right)$$

où : [D] est la diagonale de [K]

[E] est la partie triangulaire strictement inférieure de [K]

ω est un facteur de relaxation $\omega \in]0,1[$

Il s'agit donc ici de l'inverse de la matrice associée à la méthode de Sur-Relaxation Symétrique Successive (SSOR). Le grand avantage de la méthode est de ne nécessiter aucun stockage puisque [H] est directement obtenu à partir de [K]. En outre, cette méthode ne connait pas les restrictions de la précédente.

C'est donc cette méthode que nous avons décidé d'appliquer. Elle conduit à d'excellents résultats (rapidité et facilité d'emploi) sur tous les problèmes proposés.

Chapitre VII

Structures planes formées d'assemblages de poutres

Introduction

Nous nous proposerons dans ce chapitre d'étudier la déformation de structures planes formées de poutres planes chargées dans leur plan. Nous appelerons indifféremment structure ou ossature un tel assemblage de poutres.

L'étude de tels problèmes se prête bien à l'utilisation de la méthode des éléments finis, ce fut d'ailleurs là l'une des premières applications de la méthode.

Notre étude a pour but, dans un premier temps , de calculer la déformée de la structure, dans un second temps de déduire les efforts intérieurs.

Afin de mettre en évidence les performances de la méthode ainsi que ses particularités de mise en œuvre sur cette application particulière, nous avons choisi d'utiliser la méthode Full-multigrilles.

VII-1 : Notion de poutre

VII-1-1 : Définition géométrique

Une poutre droite est définie par le déplacement d'une section d'aire A dont le centre de gravité G suit une ligne droite donnée. Cette ligne droite est appelée fibre moyenne de la poutre. A tous moment, nous imposons à cette ligne d'être normale à la section.

La section de chaque poutre est repérée par un repère local direct centré sur le point G. Le premier vecteur est tangent à la fibre moyenne, les deux autres sont pris dans le plan de la section de façon à être à la fois (si cela est possible) axes d'inertie et à former un trièdre direct avec le premier vecteur. Nous supposerons par la suite que cela et toujours le cas.

Dans le repère local, le déplacement du point G est repéré par 3 degrés de libertés que nous noterons u , w et θ qui sont respectivement les déplacements suivant x et z et la rotation suivant y.

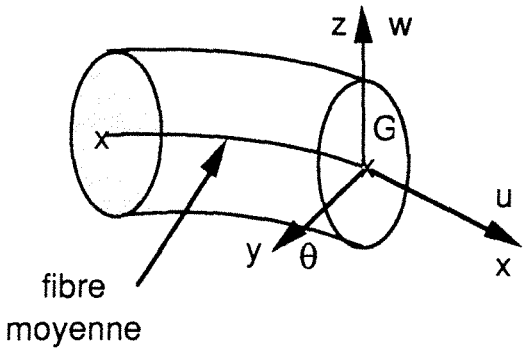


Figure VII-1 : Repère local (cas d'une poutre quelconque)

De plus , un repère global permet de repérer la position des poutres les unes par rapport aux autres.

VII-1-2 : Hypothèses générales de travail

Nous retenons l'ensemble des hypothèses suivantes :

_ Le plan Gxz est plan principal d'inertie

_ Les sections restent droites après déformation. Cependant, selon le cas, après déformation la section n'est plus forcément normale à la déformée de la fibre neutre. Ceci dépendra de l'hypothèse que nous ferons sur l'influence du cisaillement transverse, comme nous le verrons plus loin.

_ Nous ferons l'hypothèse des petites déformations. Nous pourrons alors confondre à tout moment la configuration déformée $\Gamma(t)$ avec la configuration initiale $\Gamma(o)$. Nous utiliserons donc le tenseur des contraintes de Cauchy-Euler.

_ Nous ferons l'hypothèse des contraintes planes (σ_{zz}=0) et nous supposerons que la poutre de déforme dans le plan xz

_ Nous supposerons les matériaux de chaque poutre homogènes, isotropes et isothermes.

_ Les efforts intérieurs dans chaque section sont représentés par un torseur au centre de gravité G équivalent au torseur des forces appliquées à droite de la section. Avec les hypothèses ci-dessus, les éléments de réduction de ce torseur se limitent à N, T, M, respectivement effort normal, effort tranchant et moment fléchissant. Ceux ci traduisent les effets de membrane, de cisaillement transverse et de flexion que subit la section.

VII-1-3 : Prise en compte du cisaillement transverse (formulation de Timoshenko/Midlin)

Cette théorie constitue l'un des modèles permettant l'étude des poutres en flexion en tenant compte du cisaillement transverse. Il s'agit d'une théorie dite "au 1er ordre" puisqu'elle propose un déplacement variant linéairement suivant l'épaisseur. L'étude de la rigidité de cisaillement transverse va nécessiter l'introduction des notions de section réduite et de facteur de correction.

Nous supposerons de plus que nous pouvons découpler les effets de flexion des effets de membrane, ce qui est une hypothèse très courante.

L'aspect important de la modélisation reprend une des hypothèses précédentes que nous séparons en deux points :

- Après déformation, toute section droite reste droite
- Après déformation, toute section n'est plus nécessairement perpendiculaire à la déformée de la fibre neutre

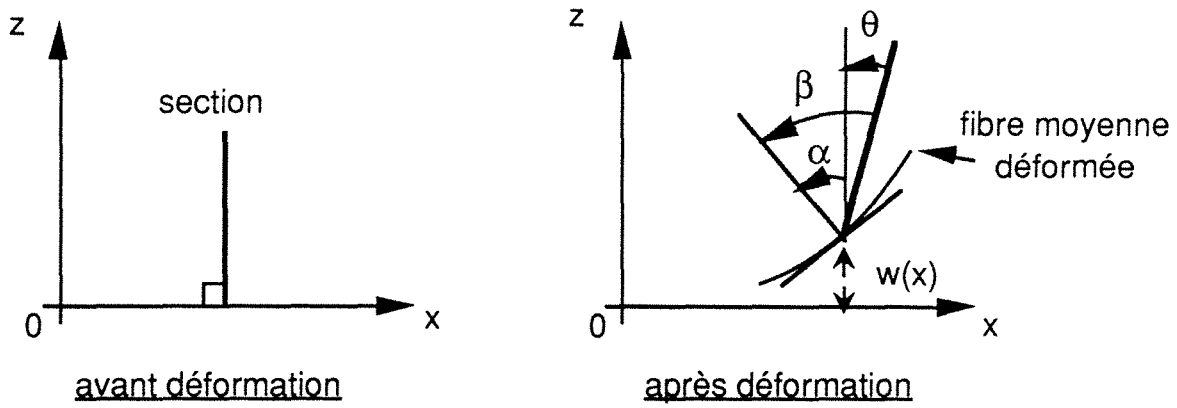


Figure VII-2 : Modélisation des sections avant et après déformation (modèle de Timoshenko)

Nous prenons donc le modèle suivant pour les déplacement :

- $u(x,z) = u(x) + z.\theta(x)$ (1)
- $w(x,z) = w(x)$ (2)
- $\theta(x) = -\frac{dw}{dx} + \beta$ (3) $\left(\alpha = \frac{dw}{dx}\right)$

L'équation (1) traduit le découplage entre effet de membrane (u(x)) et effet de flexion ($z.\theta(x)$). De plus, on notera que u(x,z) dépend linéairement de z (d'où le nom de modèle au 1er ordre).

L'équation (2) traduit le fait que tous les points de la section ont les même flèche.

Enfin, l'équation (3) montre l'effet du cisaillement transverse (angle β) qui fait que la section déformée n'est plus perpendiculaire à la déformée de la fibre moyenne.

On obtient alors les seules déformations non-nulles suivantes :

- $\varepsilon_{xx} = e + z.\chi$ • $\varepsilon_{zz} = -\nu.\varepsilon_{xx}$ (4a)

avec e = déformation de membrane = $u_{,x}$

et χ = courbure = $\theta_{,x}$

$$\bullet\ \gamma_{xz} = \frac{\partial u}{\partial z} + \frac{\partial z}{\partial x} = \theta + w_{,x} = \theta \qquad (4b)$$

avec $w_{,x}$ = rotation normale à la fibre neutre déformée

Les contraintes sont obtenues par la loi de comportement :

$$\begin{aligned} \sigma_{xx} &= E \,.\, \varepsilon_{xx} \\ \tau_{xz} &= G \,.\, \gamma_{xz} \end{aligned} \qquad (5)$$

$$\text{avec } G = \frac{E}{2.(1+\nu)}$$

Les éléments de réduction sont alors :

$$N = \int_A \sigma_{xx}\, dA \quad ; \quad T = \int_A \tau_{xx}\, dA \quad ; \quad M = \int_A z \,.\, \sigma_{xx}\, dA \qquad (6)$$

On notera qu'un tel modèle fournit une contrainte de cisaillement constante sur la section, ce qui ne vérifie pas les conditions limites (σ_{xz}=0 sur les bords supérieur et inférieur de la poutre).

Poutre à section carrée en flexion pure

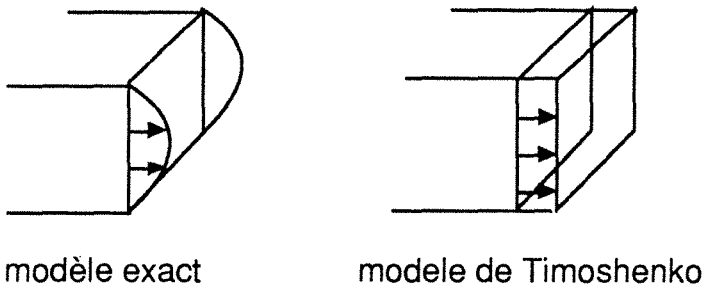


Figure VII-3 : contrainte de cisaillement selon le modèle

Pour essayer de corriger la valeur de la contrainte de cisaillement, on est amené à définir un facteur de correction k. Le calcul de k sa fait par identification des modèles de contraintes suivant un critère donné; parmi les plus courants, citons :

- Même contrainte maximale (Timoshenko)
- Même énergie de déformation (Reïsner)

Nous avons choisi le second critère. Pour les poutres pleines, la valeur de k se situe aux alentours de 1,2 . Ce facteur est utilisé pour définir une section réduite $\hat{A} = A/k$ à partir de laquelle se font les calculs de rigidité de flexion.

VII-1-4 : Modèle de poutres minces (Euler-Bernoulli)

Dans le cas de poutres minces (rapport longueur/épaisseur = L/h grand) , les relation vues en VII-1-3 peuvent se simplifier par la suppression des effets dûs au cisaillement transverse. Dès lors, même après déformation, la section reste droite et normale à la déformée de la fibre moyenne.

On obtient les relations suivantes (notations de la figure VII-4) :

- $u(x,z) = u(x) - z.\dfrac{dw}{dx}$ (7a)

- $w(x,z) = w(x)$ (7b)

- $\theta(x) = -\dfrac{dw}{dx}$ (7c)

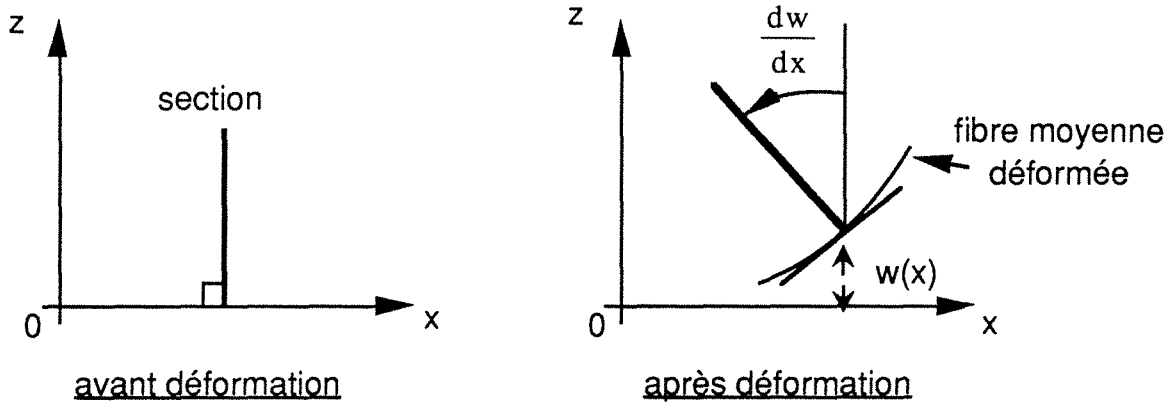


Figure VII-4 : Modélisation des sections avant et après déformation (modèle de Bernoulli)

Il reste une seule déformation non nulle donnée par :

- $\varepsilon_{xx} = u_{,x} - z.\, w_{,xx}$ (8)

Les efforts résultants sont alors données par :

$$\bullet \quad N = E.A.u_{,x} \quad ; \quad T = -E.I.\frac{d^3 w}{dx^3} \quad ; \quad M = -E.I.\frac{d^2 w}{dx^2} \qquad (9)$$

Il est alors possible d'avoir σ_{xx} par la loi de comportement. De plus , on peut évaluer la contrainte de cisaillement à partir de la connaissance de T.

VII-2 : Modélisation par éléments finis des poutres de Timoshenko

VII-2-1 : Modèle et énergie totale

Nous considérons une poutre de longueur L , d'aire A et d'inertie (par rapport à Gy) I, ces deux dernières valeurs étant constantes. Nous nous plaçons dans le repère local de la poutre (O,x,y,z).

Cette poutre est soumise aux efforts suivants :

_ des forces et des moments concentrés $\vec{F_i} = \left(F_{xi} , F_{zi} , M_{yi} \right)$

_ des forces réparties $\vec{p}(x) = \left(p_x(x) , p_z(x) \right)$ par unité de longueur

Soit la figure VII-5

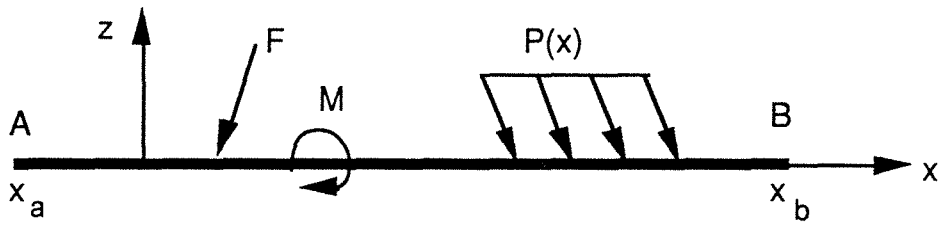


Figure VII-5 : Poutre de Timoshenko

L'energie totale de la poutre s'écrit :

$$\Pi = \int_0^L \frac{E.I}{2} \cdot \left(\frac{d\theta}{dx}\right)^2 . dx + \int_0^L \frac{G.A}{2.k} \cdot \beta^2 . dx + \int_0^L \frac{E.A}{2} \cdot \left(\frac{du}{dx}\right)^2 . dx$$

$$- \int_0^L < \vec{p} , \begin{Bmatrix} u \\ w \end{Bmatrix} > . dx - \sum_i < \vec{F_i} , \begin{Bmatrix} u \\ w \\ \theta \end{Bmatrix} > \quad (10)$$

en notant : u le déplacement suivant Ox

w le déplacement suivant Oz

θ la rotation de la section suivant Oy

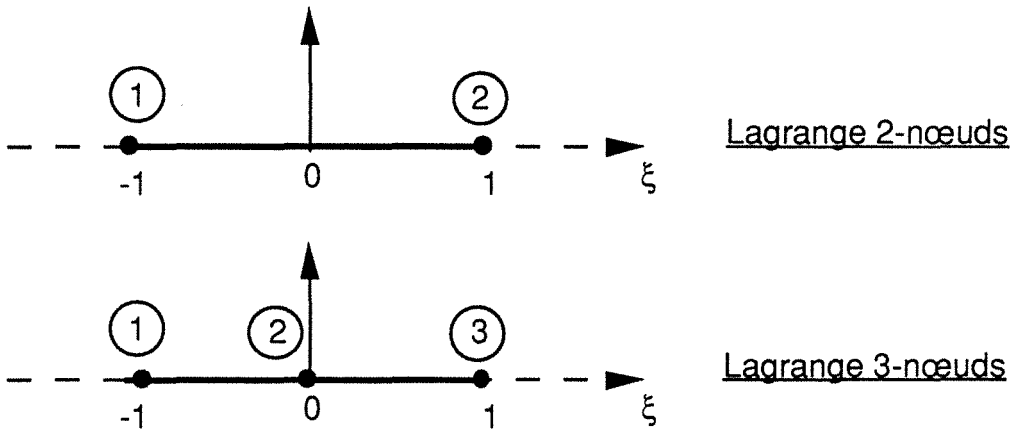
$\beta = \theta + w_{,x}$ la déformation dûe au cisaillement

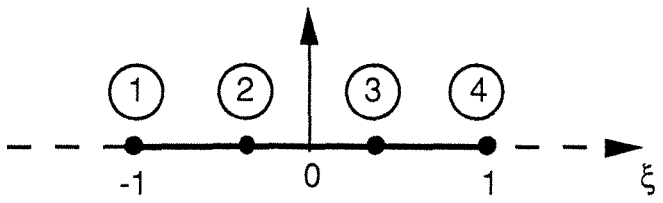
La position d'équilibre de la poutre correspond à un minimum de Π.

VII-2-2 : Choix des éléments

Nous pouvons constater que dans l'expression de Π , l'ordre maximum des dérivées ne dépasse pas 1. De ce fait, nous pouvons utiliser des éléments ayant une continuité C(0) sans risquer d'avoir des singularités aux interfaces des éléments.

Nous avons choisi pour tous les degrés de liberté, une interpolation de type Lagrange sur 2 , 3 ou 4 nœuds. Voici la représentation de ces éléments dans le repère de référence (figure VII-6) :





Lagrange 4-nœuds

Figure VII-6 : types d'éléments utilisés

Ce qui donne une interpolation du type

$$\begin{Bmatrix} u(x) \\ w(x) \\ \theta(x) \end{Bmatrix} = \sum_{i=1}^{NN} N_i(x) \cdot \begin{Bmatrix} u_i \\ w_i \\ \theta_i \end{Bmatrix} \text{ où NN= nombre de nœuds} \quad (11)$$

Ce type d'éléments présente deux inconvénients majeurs :

_ Pour certains problèmes, notamment en non-linéaire, le 2-nœuds n'est pas toujours convergent. Jusqu'à présent, en linéaire , cela n'a pas été le cas; nous gardons donc cet élément pour l'instant.

_ Lorsque l'élancement L/h croît, il se produit un phénomène de "verrouillage" [EL 3] qui provient de l'évaluation de la rigidité de cisaillement. Ce problème peut être résolu par condensation statique [EL 7] ou par une intégration réduite [EL 7] du cisaillement transverse. On effectue alors une intégration sélective (exacte pour la flexion pure et les effets de membrane, réduite pour le cisaillement transverse). Dans notre programme, c'est l'intégration sélective que nous choisirons pour remédier au problème du verrouillage.

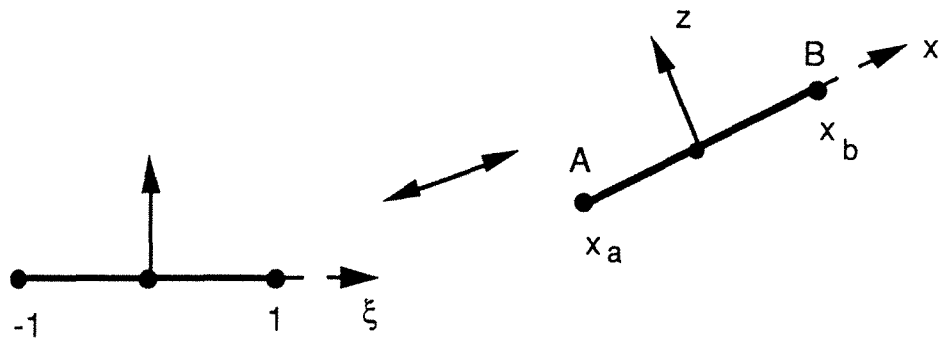
Dans le repère local, les déplacement u et w sont découplés. Notre calcul de matrice de rigidité va donc se décomposer en deux parties :

_ calcul de la rigidité de membrane (paragraphe VII-2-3).

_ calcul de la rigidité de flexion et de cisaillement (paragraphe VII-2-4).

Dans toute la suite, nous désignerons par NN le nombre de nœuds par éléments (ici, NN=2,3 ou 4).

Nous supposerons que tous les éléments réels ne sont que des transformations simples de l'élément de référence (translation, rotation, dilatation) conformément à la figure VII-7.



Elément de référence de longueur 2

Elément réel de longueur L

Figure VII-7 : Elément de référence et élément réel

VII-2-3 : Calcul de la matrice de rigidité d'une poutre en traction-compression

Nous avons par définition d'un élément isoparamétrique :

$$\begin{cases} x(\xi) = \sum_{i=1}^{NN} N_i(\xi) \cdot x_i \\ u(\xi) = \sum_{i=1}^{NN} N_i(\xi) \cdot u_i \end{cases} \qquad (11')$$

Ce que nous pouvons écrire pour la seconde équation sous la forme

$$u(\xi) = \langle N_1(\xi), \ldots, N_{nn}(\xi) \rangle \cdot \begin{Bmatrix} u_1 \\ | \\ u_{nn} \end{Bmatrix} \qquad (12)$$

D'où

$$\frac{du}{dx} = \left\langle \frac{dN_1}{dx}, \ldots, \frac{dN_{nn}}{dx} \right\rangle . \begin{Bmatrix} u_1 \\ | \\ u_{nn} \end{Bmatrix} = \left\langle B_1, \ldots, B_{nn} \right\rangle . \begin{Bmatrix} u_1 \\ | \\ u_{nn} \end{Bmatrix} \qquad (13a)$$

or

$$\frac{dN_j}{dx} = \frac{dN_j}{d\xi} . \frac{d\xi}{dx} \quad \text{avec} \quad \frac{d\xi}{dx} = \frac{2}{L} \qquad (13b)$$

Si l'on reporte cela dans l'expression de l'énergie totale de la poutre en traction-compression, on obtient alors :

$$\Pi = \frac{1}{2} . \int_{xa}^{xb} EA . \left(\frac{du}{dx}\right)^2 .dx - \int_{xa}^{xb} p_x(x).u..dx \qquad (14a)$$

qui devient :

$$\Pi = \frac{1}{2} . \int_{-1}^{1} \left\langle u_1, \ldots, u_{nn} \right\rangle . \begin{Bmatrix} B_1 \\ | \\ B_{nn} \end{Bmatrix} . [D] . \left\langle B_1, \ldots, B_{nn} \right\rangle . \begin{Bmatrix} u_1 \\ | \\ u_{nn} \end{Bmatrix} . \frac{L}{2} .d\xi$$

$$- \int_{-1}^{1} \left\langle u_1, \ldots, u_{nn} \right\rangle . \begin{Bmatrix} N_1 \\ | \\ N_{nn} \end{Bmatrix} . p_x(x) . \frac{L}{2} .d\xi \qquad (14b)$$

ou encore

$$\Pi = \frac{1}{2} . \{u\}^t . [K] . \{u\} - \{u\}^t . \{f\} \qquad (14c)$$

avec

$$\{u\} = \begin{Bmatrix} u_1 \\ | \\ u_{nn} \end{Bmatrix} \quad ; \quad \{B\} = \begin{Bmatrix} B_1 \\ | \\ B_{nn} \end{Bmatrix} \quad ; \quad \{N\} = \begin{Bmatrix} N_1 \\ | \\ N_{nn} \end{Bmatrix} \quad ; \quad [D] = [E.A]$$

$$[K] = \int_{-1}^{1} \{B\}^t . [D] . \{B\} . \frac{L}{2} . d\xi \quad \text{et} \quad \{f\} = \int_{-1}^{1} \{N\} . p_x(x) . \frac{L}{2} . d\xi$$

L'extremum de l'énergie , s'il existe, est obtenu par :

$$\frac{\partial \Pi}{\partial u_i} = 0 \quad \text{pour } i=1,..,nn \qquad (15a)$$

Ce qui conduit à $[K].\{u\} - \{f\} = 0$, c'est à dire à résoudre le système linéaire :

$$[K].\{u\} = \{f\}. \qquad (15b)$$

VII-2-4 : Calcul de la rigidité d'une poutre en flexion

Nous avons maintenant :

$$w(\xi) = \langle N_1 , .., N_{nn} \rangle . \begin{Bmatrix} W_1 \\ | \\ W_{nn} \end{Bmatrix} = \langle N \rangle . \{w\} \qquad (16a)$$

$$\theta(\xi) = \langle N_1 , .., N_{nn} \rangle . \begin{Bmatrix} \theta_1 \\ | \\ \theta_{nn} \end{Bmatrix} = \langle N \rangle . \{\theta\} \qquad (16b)$$

La déformation de flexion peut s'écrire comme étant :

$$\bullet \quad \varepsilon_{flex} = \frac{d\theta}{dx} = \left\langle \frac{dN_1}{dx} , ..., \frac{dN_{nn}}{dx} \right\rangle . \begin{Bmatrix} \theta_1 \\ | \\ \theta_{nn} \end{Bmatrix} \qquad (17)$$

La déformation de cisaillement transverse s'écrit :

$$\bullet \quad \varepsilon_{cis} = \frac{dw}{dx} + \theta = \beta \qquad \text{soit encore}$$

$$\bullet \quad \varepsilon_{cis} = \left\langle \frac{dN_1}{dx}, \ldots, \frac{dN_{nn}}{dx} \right\rangle \cdot \begin{Bmatrix} w_1 \\ | \\ w_{nn} \end{Bmatrix} + \left\langle N_1, \ldots, N_{nn} \right\rangle \cdot \begin{Bmatrix} \theta_1 \\ | \\ \theta_{nn} \end{Bmatrix} \qquad (18)$$

Et comme précédemment, on à

$$\frac{dN_i}{dx} = \frac{dN_i}{d\xi} \cdot \frac{d\xi}{dx} \quad \text{et} \quad \frac{d\xi}{dx} = \frac{2}{L} \qquad (19)$$

si nous notons $\{a\} = \begin{Bmatrix} w_1 \\ \theta_1 \\ | \\ w_{nn} \\ \theta_{nn} \end{Bmatrix}$ (20)

alors, on peut résumer les formules (17) et (18) par :

$$\begin{cases} \varepsilon_{flex} = \langle B_{flex} \rangle \cdot \{a\} \\ \varepsilon_{cis} = \langle B_{cis} \rangle \cdot \{a\} \end{cases} \qquad (21a)$$

où :

$$\langle B_{flex} \rangle = \left\langle 0 \ , \frac{dN_1}{d\xi} \cdot \frac{2}{L}, \ldots, 0 \ , \frac{dN_{nn}}{d\xi} \cdot \frac{2}{L} \right\rangle \qquad (21b)$$

$$\langle B_{cis} \rangle = \left\langle \frac{dN_1}{d\xi} \cdot \frac{2}{L}, N_1, \ldots, \frac{dN_{nn}}{d\xi} \cdot \frac{2}{L}, N_{nn} \right\rangle \qquad (21c)$$

Revenons alors à l'énergie de déformation de flexion qui vaut :

$$\Pi = \frac{1}{2} . \int_{xa}^{xb} E.I . \left(\frac{d\theta}{dx}\right)^2 . dx + \frac{1}{2} . \int_{xa}^{xb} \frac{G.A}{k} . \beta^2 . dx - \int_{xa}^{xb} p_z(x) . w . dx \qquad (22)$$

L'introduction des relations (16) à (21) conduit à une réécriture de la formule (22) ci-dessus sous forme matricielle pour obtenir:

$$\Pi = \frac{1}{2} . \{a\}^t . [K_{flex}] . \{a\} + \frac{1}{2} . \{a\}^t . [K_{cis}] . \{a\} - \{a\}^t . \{w\} \qquad (23)$$

Ceci fait apparaître deux rigidités :

- la rigidité de flexion $[K_{flex}] = \int_{-1}^{1} \{B_{flex}\}^t . [D_{flex}] . \{B_{flex}\} . \frac{L}{2} . d\xi \qquad (23b)$

Avec : $\{B_{flex}\}$ mentionné plus haut
et $[D_{flex}] = [E.I]$

- la rigidité de cisaillement $[K_{cis}] = \int_{-1}^{1} \{B_{cis}\}^t . [D_{cis}] . \{B_{cis}\} . \frac{L}{2} . d\xi \qquad (23c)$

Avec : $\{B_{cis}\}$ mentionné plus haut
et $[D_{cis}] = [G.A/k]$

- à ceci s'ajoutent les forces $\{f\} = \int_{-1}^{+1} \{N\} . p_z(\xi) . \frac{L}{2} . d\xi \qquad (23d)$

L'extremum de l'énergie , s'il existe, est obtenu par :

$$\frac{\partial \Pi}{\partial a_i} = 0 \quad i=1,..,2.nn \qquad (24a)$$

Ce qui conduit au système

$$([K_{cis}] + [K_{flex}]).\{a\} = \{f\}. \qquad (24b)$$

VII-2-5 : Calcul des diverses intégrales

Toutes les intégrales intervenant sont intégrées au moyen de quadratures de Gauss-Legendre.

Ces valeurs sont approchées de façon exacte sauf pour le cisaillement transverse qui est sous-intégré (intégration sélective).

VII-3 : Modélisation par éléments finis des poutres de Bernoulli

VII-3-1 : Modèle et énergie totale

Comme précédemment, nous considérons une poutre de longueur L, d'inertie I et d'aire A, ces deux dernières valeurs étant constantes.

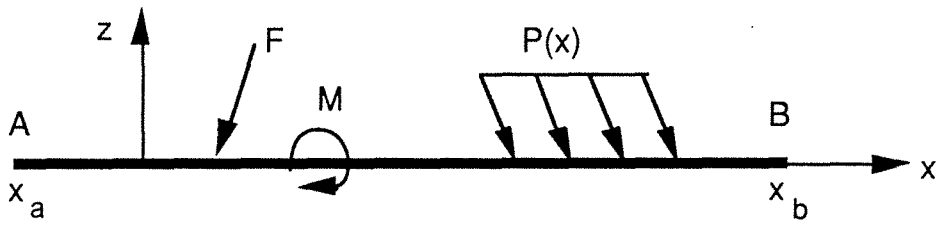


Figure VII-8 : Poutre de Bernoulli

En reprenant les notations du paragraphe VII-2, l'énergie totale de la poutre s'écrit alors :

$$\Pi = \frac{1}{2} \cdot \int_{xa}^{xb} E.I. \left(\frac{d^2 w}{dx^2} \right)^2 . dx + \frac{1}{2} \cdot \int_{xa}^{xb} E.A. \left(\frac{du}{dx} \right)^2 . dx \cdot$$

$$-\int_{xa}^{xb} < \vec{p}(x) , \begin{Bmatrix} u \\ w \end{Bmatrix} > .dx \quad - \sum_i < \vec{F}_i , \begin{Bmatrix} u \\ w \\ \theta \end{Bmatrix} > \qquad (25)$$

VII-3-2 : Choix des éléments

La présence du terme en $\frac{d^2 w}{dx^2}$ impose (au moins pour l'étude de la flexion) d'utiliser des éléments de classe C(1) pour éviter les singularités lors du calcul des intégrales [EL 3].

Cette précaution n'est pas nécessaire pour le calcul de la rigidité de membrane. Donc, en utilisant le découplage membrane-flexion, nous allons mettre en œuvre deux approximations différentes, une pour chaque phénomène.

• Traction-compression

Utilisation d'un élément de Lagrange à 2-nœuds (continuité C(0))

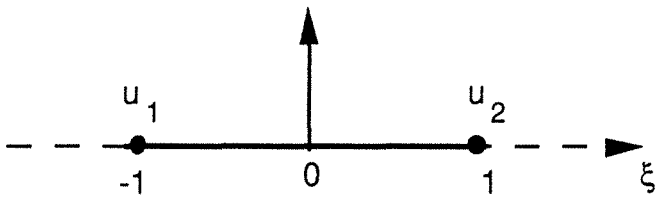


Figure VII-9 : Lagrange 2-nœuds (fonctions linéaires)

L'interpolation s'écrit pour $\xi \in [0,1]$

$$u(\xi) = N_1(\xi).u_1 + N_2(\xi).u_2 \qquad (26a)$$

$$\text{avec } N_i(\xi_j) = \delta_{ij} \quad \text{où} \quad \xi_j = \pm 1 \text{ pour } j=1,2 \qquad (26b)$$

• flexion

Utilisation d'un élément de Hermite à 2-nœuds (continuité C(1))

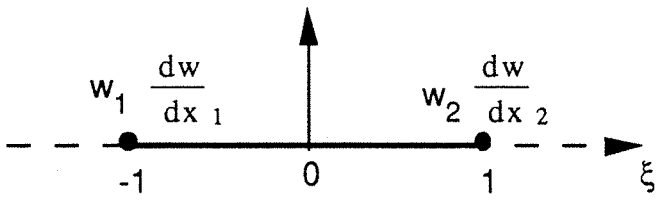


Figure VII-10 : Hermite 2-nœuds (fonctions cubiques)

L'interpolation s'écrit pour $\xi \in [0,1]$

$$w(\xi) = N_1(\xi).w_1 + \overline{N}_1(\xi).\left(\frac{dw}{dx}\right)_1 + N_2(\xi).w_2 + \overline{N}_2(\xi).\left(\frac{dw}{dx}\right)_2 \qquad (27a)$$

Les fonctions N_i et $\overline{N}_i$ sont des polynômes de degrés 3 vérifiant

$$\begin{cases} N_i(\xi_j) = \delta_{ij} & N_i'(\xi_j) = 0 \\ \overline{N}_i(\xi_j) = 0 & \overline{N}_i'(\xi_j) = \delta_{ij} \end{cases} \qquad (27b)$$

Les valeurs des fonction N_i de Lagrange et d'Hermite sont données en annexe 4.

VII-3-3 : Calcul de la rigidité d'une poutre en traction-compression

Le calcul est identique à celui du paragraphe VII-2-3.

$$u = \langle N_1 , N_2 \rangle \begin{Bmatrix} u_1 \\ u_1 \end{Bmatrix} \qquad (28)$$

nous en déduisons :

$$\frac{du}{dx} = \left\langle \frac{dN_1}{dx} , \frac{dN_2}{dx} \right\rangle \begin{Bmatrix} u_1 \\ u_1 \end{Bmatrix} = \langle B_1 , B_2 \rangle \begin{Bmatrix} u_1 \\ u_1 \end{Bmatrix} \qquad (29a)$$

avec

$$B_i = \frac{dN_i}{dx} = \frac{dN_i}{d\xi} \cdot \frac{d\xi}{dx} = \frac{dN_i}{d\xi} \cdot \frac{2}{L} \qquad i=1,2 \qquad (29b)$$

L'écriture du minimum de l'énergie totale conduit à un système linéaire $[K].\{u\} = \{f\}$ (30a)

où :

$$[K] = \int_{-1}^{1} \{B\}^t . [D] . \{B\} . \frac{L}{2} . d\xi \quad ; \quad \{f\} = \int_{-1}^{1} \begin{Bmatrix} N_1 \\ N_2 \end{Bmatrix} . p_x(x) . \frac{L}{2} . d\xi \qquad (30b)$$

et [D] = [E.A]

Dans notre cas, il n'y a qu'un seul type d'élément à considérer. Le calcul des intégrales est aisé et conduit à une formulation explicite. On trouve :

- $$[K] = \frac{E.A}{L} . \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad (31a)$$

et dans le cas où $p_x(x)$ est constant et vaut p

- $$\{f\} = \frac{p.L}{2} . \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \qquad (31b)$$

VII-3-4 : Calcul de la rigidité d'une poutre en flexion

L'énergie totale est dans ce cas :

$$\Pi = \frac{1}{2} . \int_0^L E.I . \left(\frac{d^2 w}{dx^2} \right)^2 . dx - \int_0^L p_z(x) . w(x) . dx \qquad (32)$$

Il est nécessaire de calculer la dérivée seconde de w. Nous avons :

$$w = \left\langle N_1 , \bar{N}_1 , N_2 , \bar{N}_2 \right\rangle . \begin{Bmatrix} w_1 \\ \left(dw/dx\right)_1 \\ w_2 \\ \left(dw/dx\right)_2 \end{Bmatrix} = \left\langle N_1 , \bar{N}_1 , N_2 , \bar{N}_2 \right\rangle . \{a\} \qquad (33)$$

$$\text{avec} \quad \{a\} = \left\langle \; w_1 \; \left(\frac{dw}{dx}\right)_1 \; w_2 \; \left(\frac{dw}{dx}\right)_2 \right\rangle^t$$

d'où

$$\frac{dw}{dx} = \left\langle \frac{dN_1}{dx} , \frac{d\bar{N}_1}{dx} , \frac{dN_2}{dx} , \frac{d\bar{N}_2}{dx} \right\rangle . \{a\} \qquad (34a)$$

et toujours

$$\frac{d\xi}{dx} = \frac{2}{L} \qquad (34b)$$

soit :

$$\frac{dw}{dx} = \frac{2}{L} . \left\langle \frac{dN_1}{d\xi} , \frac{d\bar{N}_1}{d\xi} , \frac{dN_2}{d\xi} , \frac{d\bar{N}_2}{d\xi} \right\rangle . \{a\} \qquad (35)$$

On obtient alors en dérivant une fois l'équation (35) et en utilisant (34b) :

$$\frac{d^2 w}{dx^2} = \frac{4}{L^2} . \left\langle \frac{d^2 N_1}{d\xi^2} , \frac{d^2 \bar{N}_1}{d\xi^2} , \frac{d^2 N_2}{d\xi^2} , \frac{d^2 \bar{N}_2}{d\xi^2} \right\rangle . \{a\} = \left\langle B_{flex} \right\rangle . \{a\} \qquad (36)$$

L'ensemble des relations (33) à (36) injectées dans la formule de l'énergie totale (32) conduit à une forme :

$$\Pi = \frac{1}{2}\{a\}^t.[K].\{a\} - \{a\}^t.\{f\} \qquad (37a)$$

La nullité de la première variation de l'énergie conduit donc à un système linéaire :

$$[K].\{a\} = \{f\} \qquad (37b)$$

$$\text{où } [K] = \int_{-1}^{1} \{B_{flex}\}^t.[D].\{B_{flex}\}.\frac{L}{2}.d\xi \quad \text{et} \quad \{f\} = \int_{-1}^{1} \{N\}.p_z(\xi).\frac{L}{2}.d\xi \qquad (37c)$$

avec :

_ <B> sous la forme mentionnée en (36)

_ [D] = [E.I]

_ $\langle N \rangle = \langle N_1 , \overline{N}_1 , N_2 , \overline{N}_2 \rangle$

Dans notre cas particulier, [K] peut s'exprimer de façon explicite par :

$$[K] = \frac{E.I}{L^3}.\begin{bmatrix} 12 & -6.L & -12 & -6.L \\ & 4.L^2 & 6.L & 2.L^2 \\ \text{symétrique} & & 12 & 6.L \\ & & & 4.L^2 \end{bmatrix} \qquad (38a)$$

et si $p_z(x)$ est constant égal à p :

$$\{f\} = \begin{Bmatrix} p.L/2 \\ -p.L^2/12 \\ p.L/2 \\ p.L^2/12 \end{Bmatrix} \qquad (38b)$$

VII-4 : Assemblage des matrices de rigidité élémentaires

VII-4-1 : Passage dans le repère global

Toutes les poutres composant l'ossature sont repérées dans un même repère global permettant de les situer les unes par rapport aux autres.

Cependant, tous les calculs de rigidité élémentaires ont été effectuées dans le repère local de chaque poutre. Il convient donc de repasser dans le repère global pour avoir une formulation générale en déplacements pour toutes les poutres.

On procède comme suit :

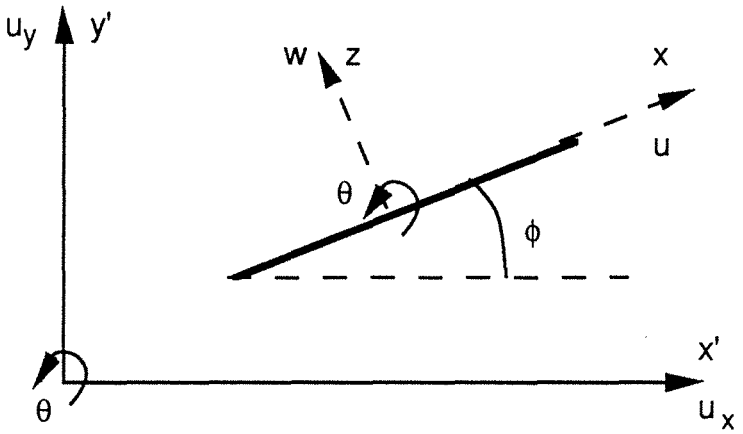


Figure VIII-11 : Repère Local et global

$\langle U_{gl} \rangle = \langle u_x , u_x , \theta_z \rangle$ est lié à $\langle U_{loc} \rangle = \langle u , w , \theta \rangle$ par

$$\{U_{gl}\} = [T] . \{U_{loc}\} \quad \text{avec} \quad [T] = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & -1 \end{bmatrix} \qquad (39)$$

La matrice de rigidité élémentaire $[K^{el}]$ totalement assemblée (membrane + flexion) est une matrice carrée d'ordre 3xNN (où NN est le nombre de nœuds par élément) atteignant au maximum 12.

La matrice élémentaire en axes globaux $\left[\overline{K}^{el}\right]$ est donnée par :

$$\left[\overline{K}^{el}\right] = [P]^t . \left[K^{el}\right] . [P] \qquad (40a)$$

où [P] est une matrice carrée d'ordre 3xNN formée des bloc 3x3 suivants :

$$[P] = \begin{bmatrix} [T] & [0] & [0] \\ [0] & & [0] \\ [0] & [0] & [T] \end{bmatrix}$$

VII-4-2 : Assemblage dans le repère global

La première phase du processus d'assemblage est le calcul pour chaque élément de la matrice élémentaire de rigidité puis sa projection dans le repère global (équation (40)).

La seconde phase est l'assemblage à proprement parler dans la matrice globale. La matrice globale du système traduit le comportement de l'ensemble de la structure, cette structure étant la "somme" de ses éléments composants. Ce dernier aspect se retrouve dans la matrice globale qui est la somme des termes de chaque matrice élémentaire dispatchés à une place judicieuse. Cette simple "somme" s'explique par la linéarité des opérateurs intervenant.

Le dispatching des termes est effectué de la façon suivante :

> La numérotation des nœuds de chaque élément de 1 à NN est purement locale et utilisée dans l'élément de référence. Dans l'élément réel, les véritables numéros des nœuds sont N(1),...,N(NN) (figure VII-12).

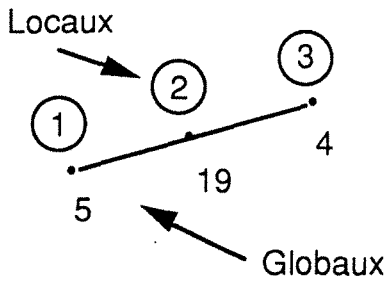


Figure VII-12 : Exemple de numérotation (élément 3-nœuds)

La matrice de rigidité élémentaire (en axes globaux) $\left[\overline{\overline{K}}^{el}\right]$ a une structure de blocs 3x3. Chaque bloc rassemble les 3 degrés de liberté globaux (u_x,u_y,θ) communs à chaque nœud.

$$\left[\overline{\overline{K}}^{el}\right] = \begin{bmatrix} \left[\overline{\overline{K}}^{el}_{1\ 1}\right] & \cdots & \left[\overline{\overline{K}}^{el}_{1\ NN}\right] \\ \vdots & & \vdots \\ \left[\overline{\overline{K}}^{el}_{NN\ 1}\right] & \cdots & \left[\overline{\overline{K}}^{el}_{NN\ NN}\right] \end{bmatrix}$$

Cette structure bloc se retrouve dans la matrice globale de l'ensemble de la de la structure qui se trouve formée de blocs 3x3 $\left[\overline{\overline{K}}^{el}_{ij}\right]$.

La formule d'assemblage est alors :

$$\left[K^{gl}_{N(i)\ N(j)}\right] = \left[K^{gl}_{N(i)\ N(j)}\right] + \left[\overline{\overline{K}}^{el}_{ij}\right] \qquad (41)$$

Cette procédure est appliquée à l'ensemble des éléments de la structure pour donner la totalité de la matrice de rigidité de l'ossature.

Après avoir vu le détail de la modélisation, nous allons maintenant décrire l'architecture du logiciel permettant de résoudre les problèmes de poutres planes.

VII-5 : Pré-processeur d'entrée des données

Les modules interactifs de saisie des données sont écrits en FORTRAN 77 et utilisent la TOOLBOX MACINTOSH comme intermédiaire graphique. Ils sont gérés par menus déroulants et par saisie directe à l'écran des données ou des entités à affecter.

La structure du programme, tant en gestion de données qu'en résolution a été concue de façon modulaire afin de pouvoir aisément faire des ajouts ou des modifications de programme.

VII-5-1 : Entrée de la géométrie de la structure

Elle se décompose en trois étapes :

_ a) entrée des points-clés

_ b) entrée des éléments-clés

_ c) maillage de la structure

a) • Les points-clés sont les points remarquables de la structure (appuis, charges, changement de type de poutre, etc) sur lesquels s'appuie le maillage.

Ces points sont entrés par l'utilisateur de façon interactive. L'écran graphique s'adapte automatiquement à la taille de l'ossature créée.

Des utilitaires (symétrie, translation, ..., mais aussi destruction) facilitent l'entré de ces points en la rendant plus souple d'emploi.

b)• Les éléments-clés s'appuient obligatoirement sur les points-clés ; Leur construction se fait automatiquement par la saisie de deux points-clés à l'écran. Un utilitaire permet la construction automatique de lignes polygonales construites de proche en proche, option très utile pour les treillis. Un autre utilitaire permet de construire des arcs de cercle (qui seront approchés par des poutres cependant). Enfin, il est possible de détruire à la demande des éléments-clés jugés incorrects.

Le logiciel permet la combinaison de ces trois options dans n'importe quel ordre.

A ce niveau, il est possible de stocker les éléments-clés et les points en fichier, ce qui permettra de reprendre ultérieurement l'étude.

Le choix du type d'éléments, leur densité, les matériaux, les forces et les blocages seront alors à redéfinir. Seule la géométrie des éléments-clés est conservée.

Une fois cette géométrie rappelée, il est possible de la modifier par les trois utilitaires vus ci-dessus.

c) • Dans un premier temps, l'utilisateur choisit le type d'élément qu'il souhaite utiliser (Bernoulli 2-nœuds ou Timoshenko 2-, 3- ou 4-nœuds). Il choisit ensuite une densité globale (fin, moyen,...etc) selon laquelle la structure est découpée. Il est alors possible de changer à volonté la densité de division d'un ou de plusieurs éléments-clés par simple saisie à l'écran

La division de chaque élément-clé se fait de façon linéaire. On peut cependant aussi diviser de façon logarithmique , ce qui permet de concentrer les éléments sur l'un ou l'autre des points-clés définissant l'élément.

Une fois ceci défini, le maillage se réalise automatiquement dans le type d'élément choisi.

VII-5-2 : Affectation des matériaux

Le choix des différents matériaux est effectué une fois le maillage réalisé par saisie directe à l'écran. Pour chaque type de poutre, on rentre :

_ Le module d'Young E

_ Le coefficient de Poisson ν

_ La masse volumique ρ

_ L'inertie principale de flexion I

_ L'aire de la section A

_ L'épaisseur de la poutre h

_ Le coefficient de correction k (si besoin est)

_ Le moment statique m

Ces caractéristiques sont stockées dans un fichier séquentiel à accès direct.

VII-5-3 : Prise en compte des conditions limites

Il est possible d'imposer une valeur donnée à n'importe quel degrés de liberté d'un nœud donné. Le cas le plus fréquent est celui des appuis ou l'on impose à un ou plusieurs déplacements généralisés la valeur nulle.

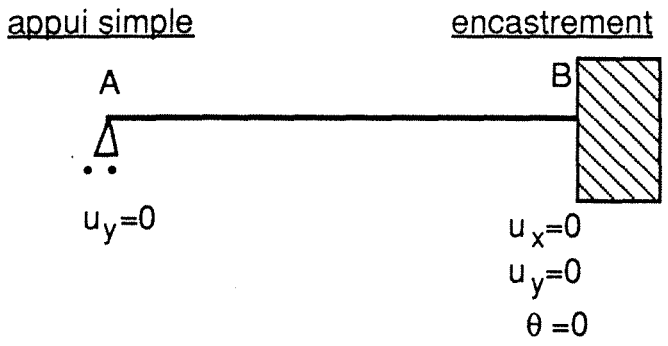


Figure VII-13 : Exemples de déplacement imposés

Il est aussi possible de traiter des cas inclinés, c'est à dire l'imposition d'un déplacement suivant une direction qui n'est pas parmi celles du repère global

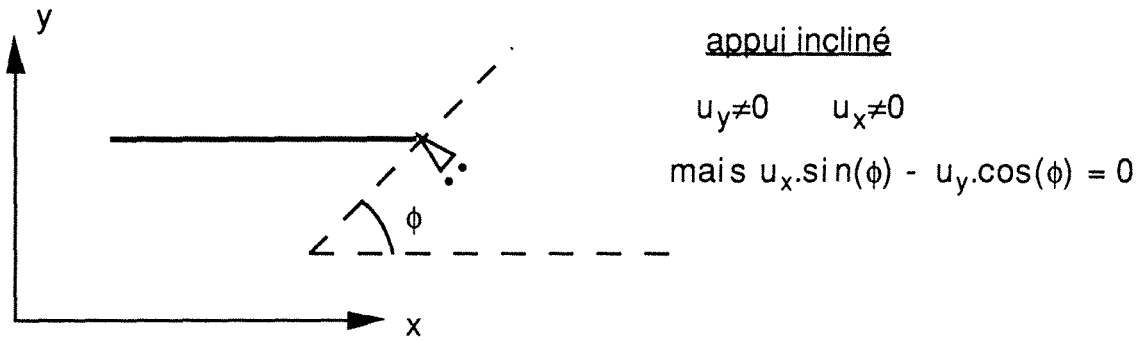


Figure VII-14 : Appui incliné

Les déplacement imposés sont saisis à l'écran puis stockés en fichier direct. Un cas particulier est prévu pour les déplacements imposés inclinés.

Pour des facilités numériques et informatiques, la prise en compte des déplacements imposés ne se fait qu'une fois la matrice de rigidité totalement assemblée. Nous utilisons la méthode de pénalisation dont voici l'explication :

Supposons que le déplacement u_i soit imposé à la valeur u_{i0}. La $i^{éme}$ équation du système avant modification s'écrit :

$$\sum_{j=1}^{n} k_{ij} \cdot u_j = f_i \qquad (42a)$$

Cette $i^{éme}$ équation doit être remplacée par l'équation $u_i = u_{i0}$. Pour cela ajoutons au terme diagonal un nombre d>0 tel que $d \gg k_{ij}$ j=1,...,n et forçons le second membre à être $(d \times u_{i0})$. On obtient :

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} k_{ij} \cdot u_j + (d + k_{ii}) \cdot u_i = u_{i0} \cdot d \qquad (42b)$$

soit

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{k_{ij}}{d} \cdot u_j + (1 + \frac{k_{ii}}{d}) \cdot u_i = u_{i0} \qquad (42c)$$

L'hypothèse $d \gg k_{ij}$ assure que numériquement, l'équation (42c) équivaut à :

$$u_i \approx u_{i0}$$

Un cas particulier est prévu pour les déplacements inclinés.

L'avantage de cette méthode est de ne pas modifier la topologie de la matrice de rigidité. Cela permet de découpler la rigidité de la structure des conditions limites et de pouvoir envisager une autre utilisation de la même ossature avec des conditions limites différentes.

actuellement, ces développements sont en cours pour inclure les conditions d'appuis élastiques.

VII-5-4 : Forces appliquées

Elle sont de trois types possible :

_ les forces ponctuelles

_ les forces linéiques

_ le poids propre

L'ensemble des forces extérieures correctement assemblé constitue le second membre global {f}.

• forces ponctuelles

$$\{f\} = \sum_{i=1}^{NFP} \{f_i\} \qquad \text{où} \begin{cases} \{f_i\} = \langle F_{ix}\ ,\ F_{iz}\ ,\ M_{ix} \rangle \\ NFP = \text{nombre de forces ponctuelles} \end{cases} \tag{43a}$$

• forces linéiques

$$\{f\} = \sum_{i=1}^{NFL} \int_{e_i} \{N\} \,.\, \rho \,.\, dx \qquad \text{où} \begin{cases} \{N\} = \text{fonctions de forme de l'élément } e_i \\ NFL = \text{nombre d'éléments } e_i \text{ chargés} \end{cases} \tag{43b}$$

Sur chaque élément, les forces linéiques sont définies par les valeurs aux deux extrémités, ceci permettant de définir des profils de forces trapézoïdaux.

Les forces peuvent être, au choix, définies dans le repère global ou dans le repère local de chaque poutre (figure VII-15).

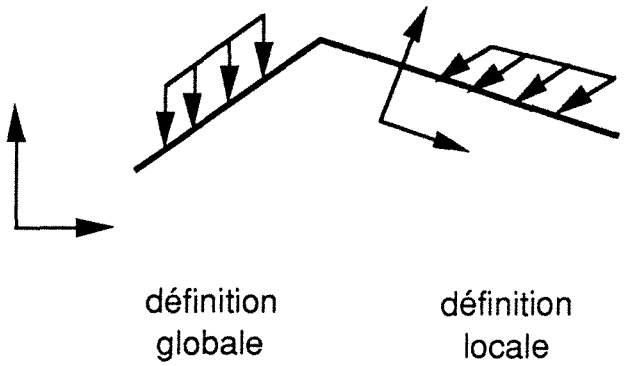


Figure VII-15 : types de forces linéiques

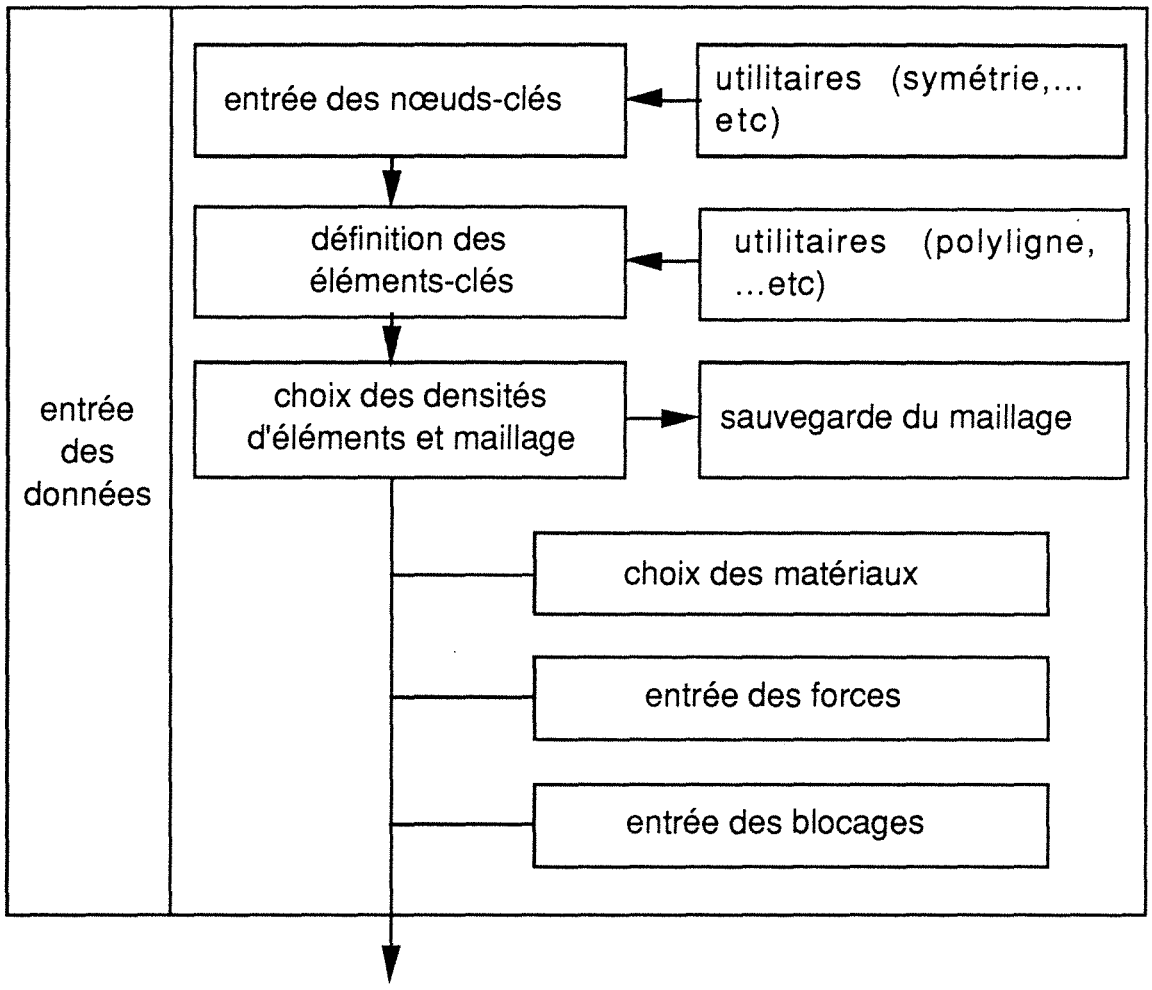
Les intégrales intervenant sont évaluées numériquement par quadratures de Gauss-Legendre.

• Poids propre

En connaissant la masse volumique ρ et l'aire A, on peut transformer le poids propre de chaque poutre en charge linéique équivalente , et de là, on se ramène au cas précédent.

VII-5-5 : Algorithme d'entrée des données

Nous pouvons résumer les phases d'entrée de l'ossature et de modélisation par la figure suivante



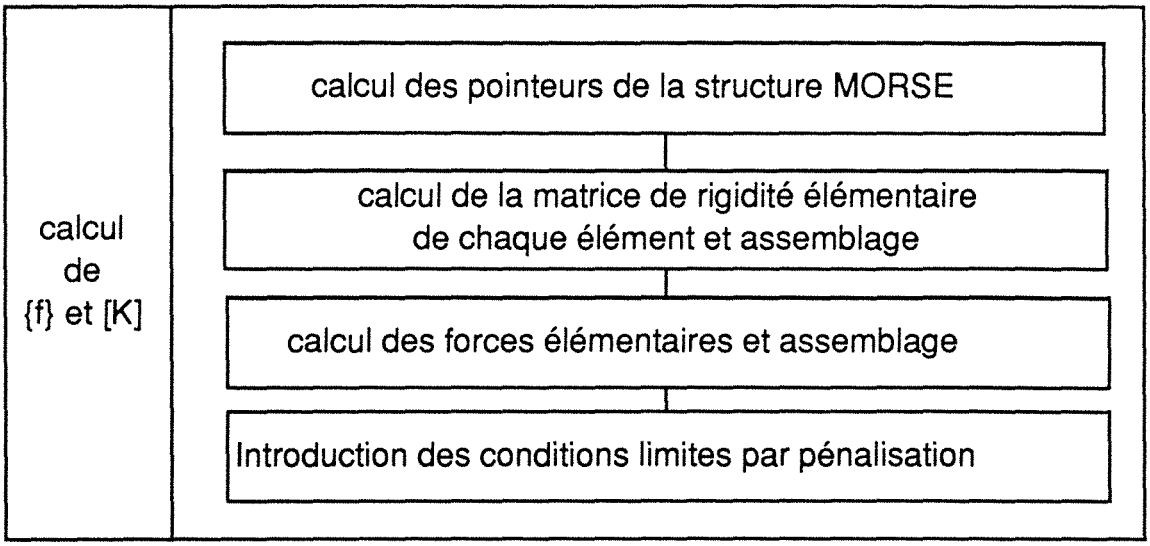


Figure VII-16 : Algorithme d'entrée des données

VII-6 : Résolution du système

La résolution du système global [K].{u} = {f} se fait par une méthode Full-multigrilles parfaitement identique à celle vue au chapitre VI .

Nous allons maintenant insister sur les particularités de la méthode dans le cas des ossatures de poutres planes .

VII-6-1 : Divisions de grilles

Dans tous les cas, la grille i est issue de la grille i-1 par dichotomie .

Grille (i-1) Grille (i)

Figure VII-17 : Exemple de divisions d'éléments

VII-6-2 : Prolongations et restrictions

La présence d'éléments C(0) et C(1) modifie quelque peu nos formules de prolongations et de restrictions . Cependant, nous prendrons toujours $r=p^t$.

• prolongation d'éléments de Bernouilli

Nous nous plaçons dans la géométrie (locale suivante)

Figure VII-18 : prolongation d'éléments de Bernouilli

Nous appelons h la taille de l'élément I sur la grille i (comme indique sur la figure VII-18) et u_i, w_i, θ_i les trois valeurs associées à chaque nœud i .

- $u'_1 = u_1 \; ; \; w'_1 = w_1 \; ; \; \theta'_1 = \theta_1$
- $u'_3 = u_2 \; ; \; w'_3 = w_2 \; ; \; \theta'_3 = \theta_2$
- $u'_2 = \dfrac{(u_1 + u_2)}{2}$
- $w'_2 = \dfrac{(w_1 + w_2)}{2} + \dfrac{h \,.\, (\theta_1 - \theta_2)}{8}$
- $\theta'_2 = -\dfrac{(\theta_1 + \theta_2)}{4} + \dfrac{3 \,.\, h \,.\, (w_2 - w_1)}{2}$

Tableau VII-1 : prolongation d'éléments de Bernouilli

• restriction d'éléments de Bernoulli

$$\bullet \quad u_1 = u'_1 + \frac{u'_2}{2} \quad ; \quad u_2 = u'_3 + \frac{u'_2}{2}$$

$$\bullet \quad w_1 = w'_1 + \frac{w'_2}{2} - \frac{3 \,.\theta'_2}{2 \,.\, h} \quad ; \quad w_2 = w'_3 + \frac{w'_2}{2} + \frac{3 \,.\theta'_2}{2 \,.\, h}$$

$$\bullet \quad \theta_1 = \theta'_1 - \frac{\theta'_2}{4} - \frac{h \,.w'_2}{2} \quad ; \quad \theta_2 = \theta'_3 - \frac{\theta'_2}{2} - \frac{h \,.w'_2}{2}$$

Tableau VII-1 : restriction d'éléments de Bernoulli

• restriction et prolongation d'éléments de Timoshenko

Nous allons donner successivement les restrictions concernant les éléments 2 , 3 et 4-nœuds. Dans tous les cas , nous prendrons $p = r^t$.

Restriction: élément de Timoshenko à 2 nœuds	 Grille (i+1) Grille i
	• $u_1 = u_1' + \frac{u_2'}{2}$; $u_2 = u_3' + \frac{u_2'}{2}$ • $w_1 = w_1' + \frac{w_2'}{2}$; $w_2 = w_3' + \frac{w_2'}{2}$ • $\theta_1 = \theta_1' + \frac{\theta_2'}{2} + \frac{h \cdot w_2'}{8}$; $\theta_2 = \theta_3' + \frac{\theta_2'}{2} - \frac{h \cdot w_2'}{8}$

Tableau VII-3 : restriction pour un élément de Timoshenko 2-nœuds

Restriction: élément de Timoshenko à 3 nœuds	 Grille (i+1) Grille i
	$u_i = u_i + \sum_{j=1}^{5} N_i(\xi_j) \cdot u_j'$; $w_i = w_i + \sum_{j=1}^{5} N_i(\xi_j) \cdot w_j'$ (i=1,3) $\theta_1 = \sum_{j=1}^{5} N_1(\xi_j).\theta_j' + \frac{h.w_2'}{16}$ $\theta_2 = \sum_{j=1}^{5} N_2(\xi_j).\theta_j' + \frac{h.(w_3' - w_2')}{16}$ $\theta_3 = \sum_{j=1}^{5} N_3(\xi_j).\theta_j' - \frac{h.w_3'}{16}$

Tableau VII-4 : restriction pour un élément de Timoshenko 3-nœuds

Restriction: élément de Timoshenko à 4 nœuds	 $u_i = u_i + \sum_{j=1}^{7} N_i(\xi_j) \cdot u_j' \quad ; \quad w_i = w_i + \sum_{j=1}^{7} N_i(\xi_j) \cdot w_j' \qquad (i=1,4)$ $\theta_1 = \sum_{j=1}^{5} N_1(\xi_j).\theta_j' + \frac{h.w_2'}{20}$ $\theta_2 = \sum_{j=1}^{5} N_2(\xi_j).\theta_j' + \frac{h.(w_4' - w_2')}{20}$ $\theta_3 = \sum_{j=1}^{5} N_3(\xi_j).\theta_j' + \frac{h.(w_6' - w_4')}{20}$ $\theta_4 = \sum_{j=1}^{5} N_4(\xi_j).\theta_j' - \frac{h.w_6'}{20}$

Tableau VII-4 : restriction pour un élément de Timoshenko 4-nœuds

Dans ces trois tableaux, les fonction N_i sont les fonctions de forme de l'élément grossier (grille i) et les ξ_i sont les positions des points de la grille fine dans le repère de référence de l'élément grossier ($\xi_i \in [0,1]$) .

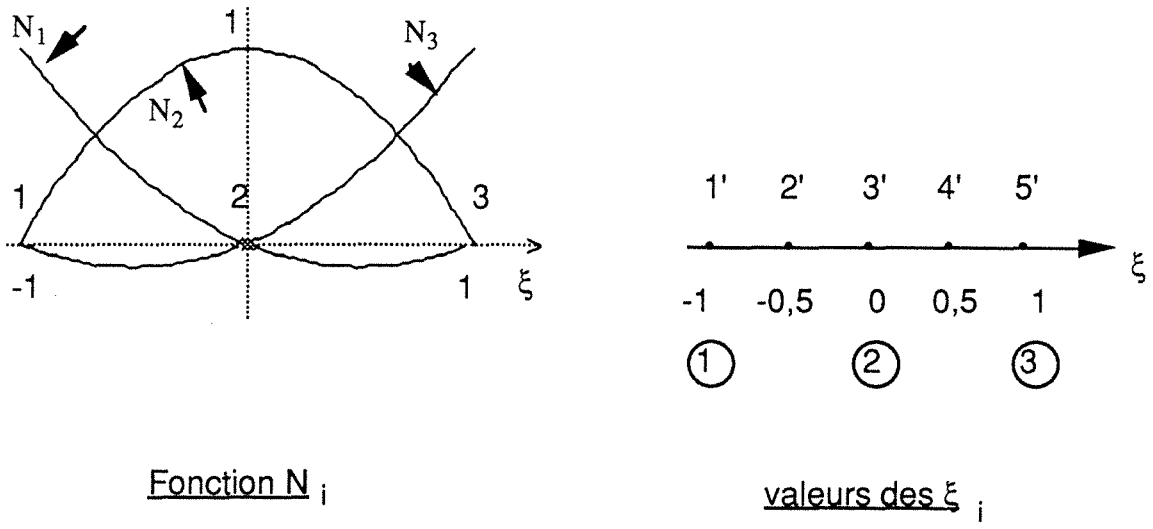


Figure VII-19 : Un exemple des fonctions N_i et des points ξ_i

• une remarque importante

Tous les relations de restriction et de prolongation sont écrites dans le système local de chaque poutre. Or, la plupart des valeurs (le défaut entre-autres) sont exprimées dans le repère global. Il faudra donc pour chaque élément, utiliser la matrice de passage

$$[T] = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad \text{ou son inverse } [T]^{-1}$$

pour passer d'un repère à l'autre.

VII-6-3 : Stockage des matrices de rigidité

La question est de savoir s'il vaut mieux recalculer les matrices de rigidité ou bien les recalculer à chaque fois. Dans l'optique de la méthode Full-multigrilles, cette seconde option est à exclure. En effet, les matrices de rigidité sont très souvent utilisées (lissage ou résolution exacte) et la fréquence de ces appels croît fortement avec le nombre de grille et le type de cycle. Par ailleurs, le type de matrice sur grille grossière choisi fait que l'on rencontre toujours la même matrice à un niveau de grille donné.

Nous avons donc choisi la première option. Toutes les matrices intervenant sont stockées selon la méthode MORSE. Elles résident toutes sur fichier disque segmenté à accès direct (comprenant la matrice et sa topologie). Lorsqu'une matrice est nécessaire, celle-ci est rappelée en mémoire centrale, intégralement ou par morceaux, de façon à améliorer le temps de calcul.

Si l'on compare notre stockage multigrilles à celui d'une méthode classique (une seule grille au niveau le plus fin), on a , d'un côté le stockage de n matrices de rigidités de plus en plus petite, et de l'autre, le stockage de la matrice au niveau le plus fin. Ceci fait que la méthode multigrilles nécessite environ entre 1,5 et 2 fois plus de place mémoire qu'une méthode classique (ceci est aussi vrai pour les vecteurs {u} et {b}). Cependant ce rapport varie selon le type de changement de grille utilisé , mais il donne quand même une bonne idée des problèmes de place mémoire auxquels on pourra être confronté.

VII-7 : Post-traitement des résultats

VII-7-1 : Visualisation de la déformée

La déformée de la structure peut être visualisée directement à l'écran. Celle ci peut être amplifiée d'un rapport donné de façon à mieux la visualiser. Des zones particulières peuvent être zoomées à la demande.

Ces deux utilitaires sont gérés par menus déroulant utilisant une saisie directe des données à l'écran.

VII-7-1 : Calcul des réactions et des efforts intérieurs

Une fois connue les déplacements nodaux, nous calculons les réactions et les efforts intérieurs.

Les efforts sont calculés dans le repère local de chaque élément par {efforts] = [K].{u}. Ces efforts sont ensuite assemblés dans le repère global pour donner les réactions.

VII-7-1 : Calcul des diagrammes de résistance des matériaux

Nous travaillons élément-clé par élément-clé. Sur chacun de ces éléments, nous avons à évaluer et à tracer les variables suivantes, toutes exprimées dans le repère local.

_ les trois déplacements généralisés : u, w, θ

_ l'effort normal : N

_ l'effort tranchant : T

_ le moment fléchissant : M

_ la contrainte de cisaillement : τ

_ les contraintes normales hautes et basses de la section : σ_{xx}

Les relations de définition de ces variables dépendent de la modélisation faite ;

Elément de poutre de Timoshenko :

Effort normal	$N = E.A.u'$
Moment fléchissant	$M = E.I.\theta'$
Effort tranchant	$T = k.G.A.\gamma_{zz} = k.G.A.\beta$
Cisaillement	$\tau = G.\beta$
contraintes normales	$\sigma_{xx} = N/A \pm M.h/I$

Tableau VII-6 : Diagrammes de résistance des matériaux pour une poutre de Timoshenko.

Elément de poutre de Bernoulli

rotation de section	$\theta = - w'$
Effort normal	$N = E.A.u'$
Moment fléchissant	$M = E.I.\theta' = - E.I.w''$
Effort tranchant	$T = - E.I\ w'''$
Cisaillement	$\tau = (T.m)/(I.h)$
contraintes normales	$\sigma_{xx} = N/A \pm M.h/I$

Tableau VII-6 : Diagrammes de résistance des matériaux pour une poutre de Bernoulli.

avec :

- _ E = module d'Young
- _ G = module de coulomb
- _ A = aire de la section
- _ I = inertie principale
- _ h= épaisseur de la poutre
- _ m = moment statique de la section

Toutes ces variables sont calculées élément-clé par élément-clé à partir des déplacements nodaux. Nous approchons les déplacements par des fonctions splines cubiques qui constituent une bonne approximation de ceux-ci. Nous en profitons pour en déduire la valeur de la dérivée sur l'élément-clé. Il est à noter cependant (voir annexe 5) que la valeur de la dérivée est généralement mauvaise aux deux extrémités de l'élément-clé. Ces deux dérivées sont alors calculées séparément.

L'utilisation des fonctions splines cubiques permet donc les calculs de résistance des matériaux. Ceux-ci sont stockés sur fichier disque à accès direct. A partir de ce fichier, il est possible de visualiser n'importe quelle courbe pour un ou plusieurs éléments-clés saisis à l'écran par l'utilisateur. La courbe tracée est automatiquement lissée par fonctions splines cubiques.

Pour chaque courbe, on spécifie les valeurs maximales atteintes ainsi que les échelles en x et y. Ces courbes peuvent être imprimées par simple recopie d'écran.

VII-8 : Exemples

VII-8-1 Poutre sur cinq appuis

Le premier exemple que nous traitons est une poutre à quatre travée sur cinq appuis et uniformément chargée (figure VII-20-a).

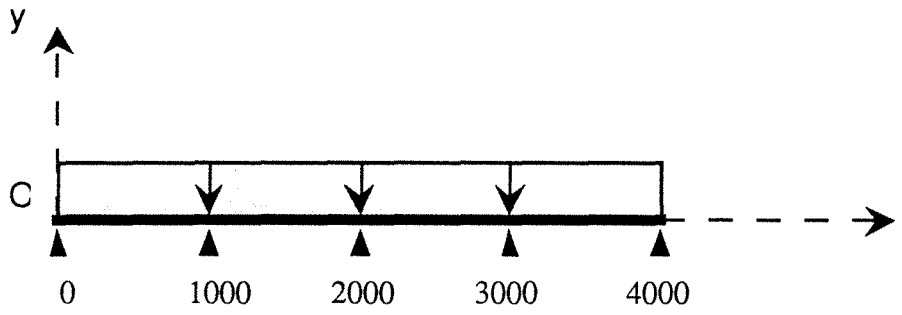


Figure VII-20-a : Géométrie du problème

L'axe Ox est pris suivant la fibre moyenne de la poutre

La poutre est une HEA 140 en acier de longueur 4000 mm. Chaque appui est disposé à 1000 mm du précédent. L'ensemble de la structure est soumise à une force linéique constante de densité p=-1000 N/mm.

Nous effectuons un processus 3-grilles w-cycle, lissage par quatre itérations d'un processus de Gauss-Seidel. A chaque niveau, nous effectuons 6 itérations de processus multigrilles. La grille la plus grossière est constituée de 8 éléments de poutre de Bernoulli (un nœud entre chaque appui, cf figure VIII-20-b), les grilles suivantes étant déterminées par dichotomie.

Figure VII-20-b : grille la plus grossière

Nous obtenons les résultats suivants :

valeur	u_y (x=500) en mm	θ_z (x=500) en radians
référence [MG 9]	-2.91541269	0.002057938
notre logiciel	-2.91545174	0.002057935

L'étude que nous avons réalisé se termine par le tracé des diagrammes de résistance des matériaux à partir des résultats obtenus sur la grille la plus fine.

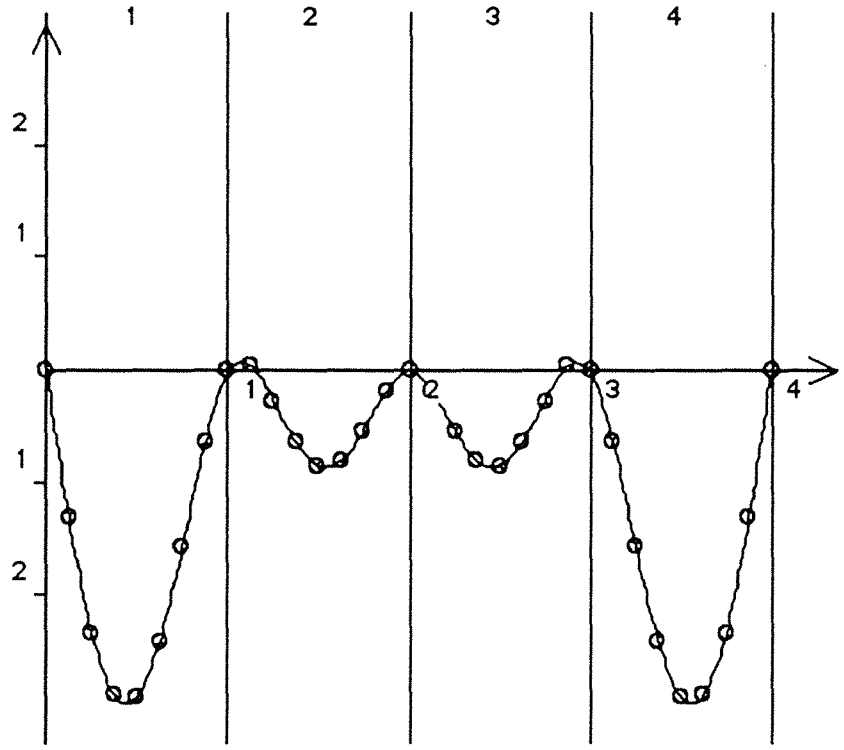


Figure VII-20-c : Tracée du déplacement selon Oy
(le maximum étant de -2.91545 mm)

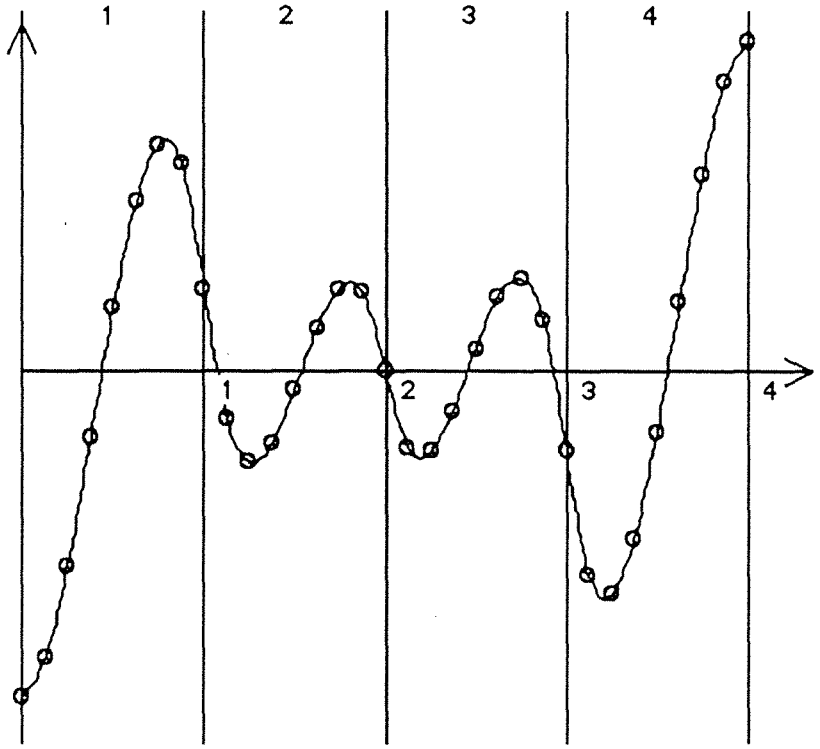


Figure VII-20-d : Tracée de la rotation de la section
(le maximum étant de 0.0011 rd)

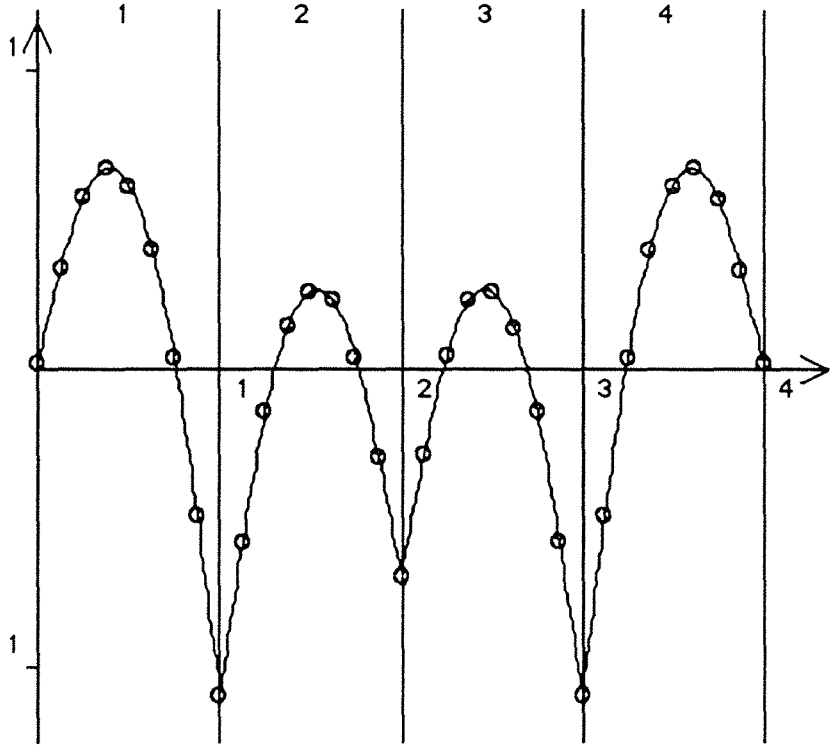


Figure VII-20-e : Tracée du moment fléchissant
(le maximum étant de 110 N.mm)

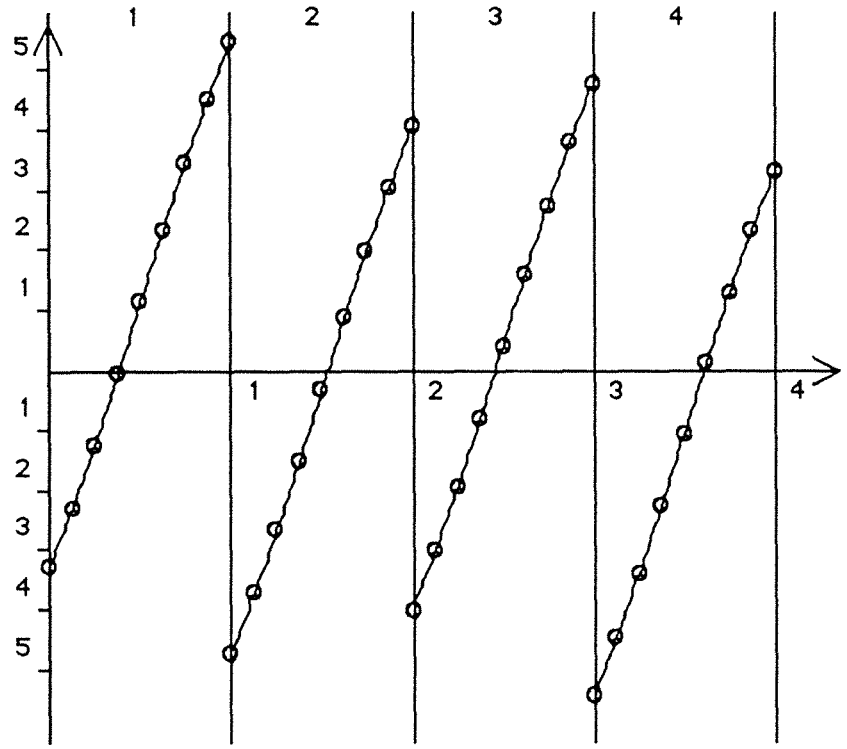


Figure VII-20-f : Tracée de l'effort tranchant (le maximum étant de 55000 N)

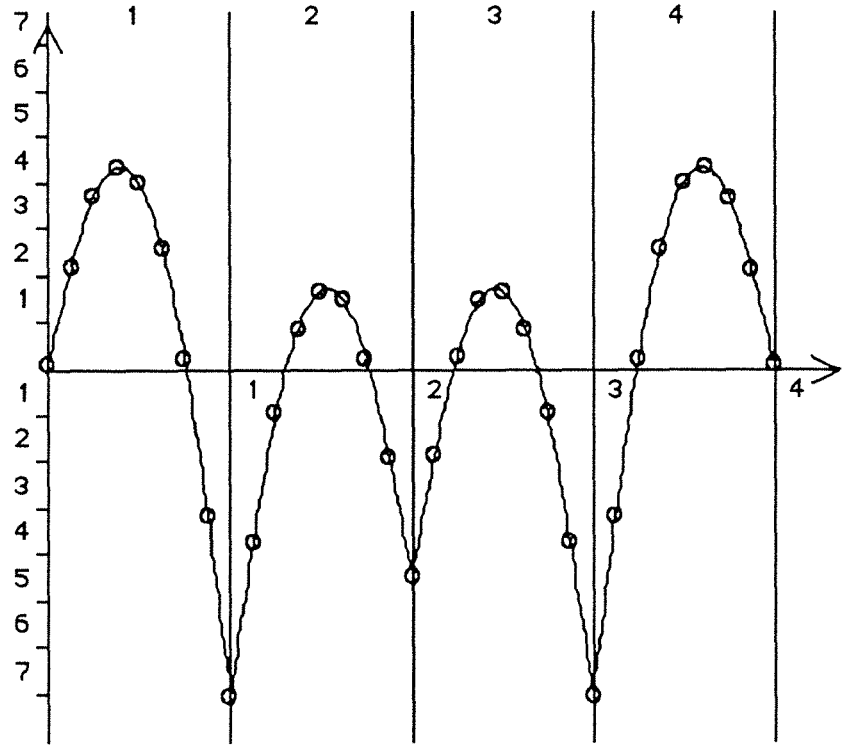


Figure VII-20-f : Tracée de la contrainte σ_{xx} haute (le maximum étant de 708 M.Pa)

VII-8-2 Poutre sur deux appuis

Nous nous intéressons au problème suivant :

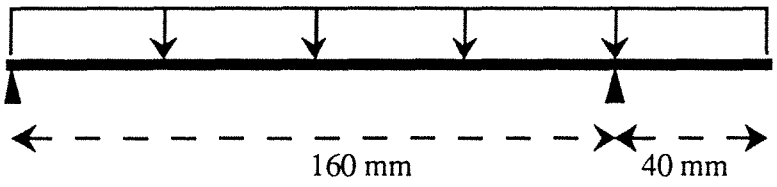


Figure VII-21 : Le problème traité

La poutre est en acier (E=210000 N/mm^2, $\nu = 0,3$) carrée de 10 mm de côté.

Nous utilisons un processus full Multigrilles à 3-grilles (4 itérations à chaque niveau). La grille la plus grossière comprend deux éléments de poutre de Bernoulli sur chaque tronçon et quatre nœuds.

Nous obtenons sur la grille la plus fine les résultats suivant:

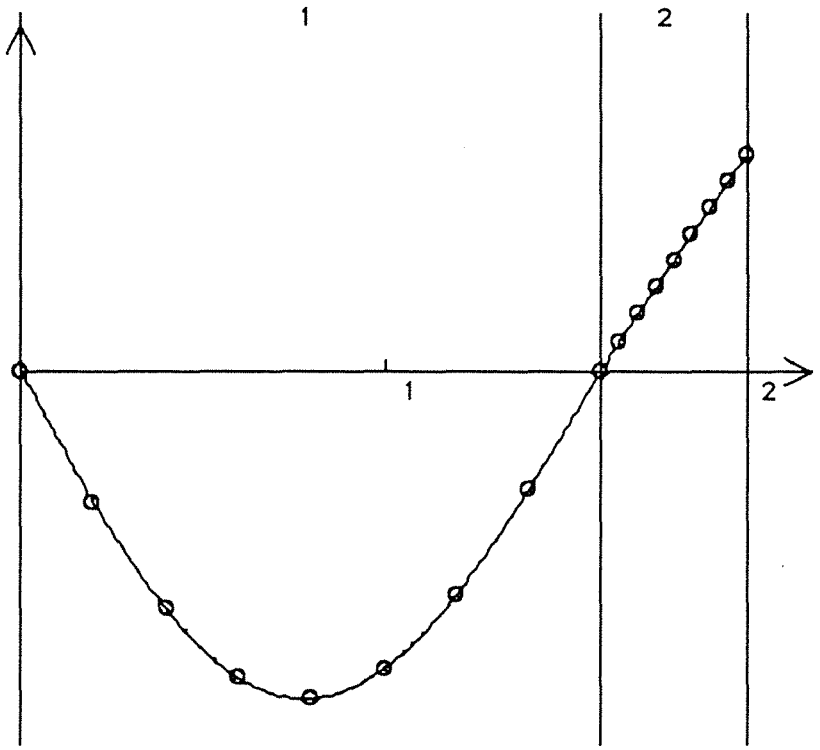


Figure VII-22-a :Tracé du déplacement v selon Oy (maximum : 0.4979 mm)

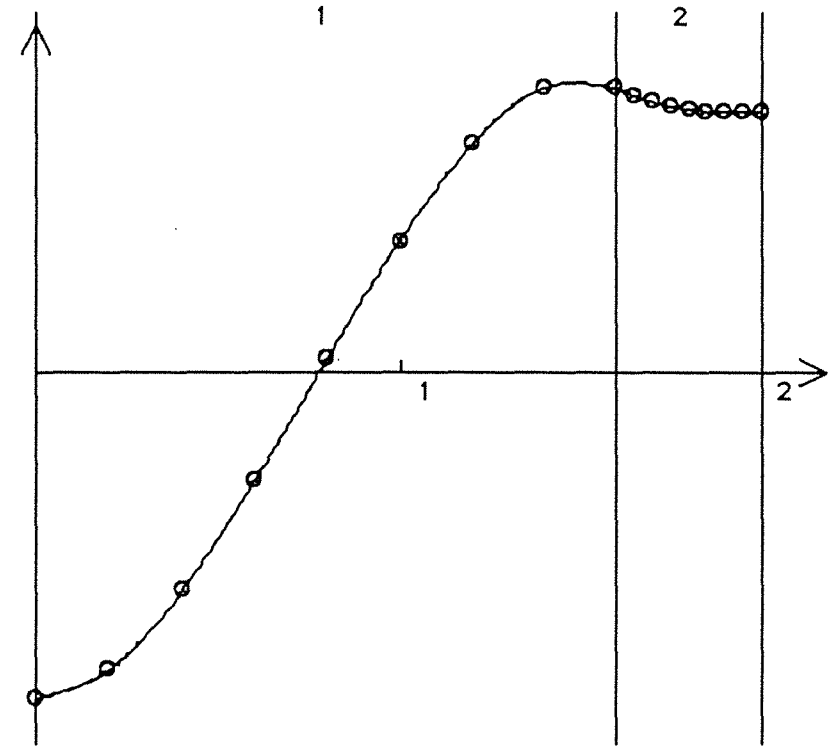


Figure VII-22-b :Tracé de la rotation de la section (le maximum : 1.02 10^{-2}-rd)

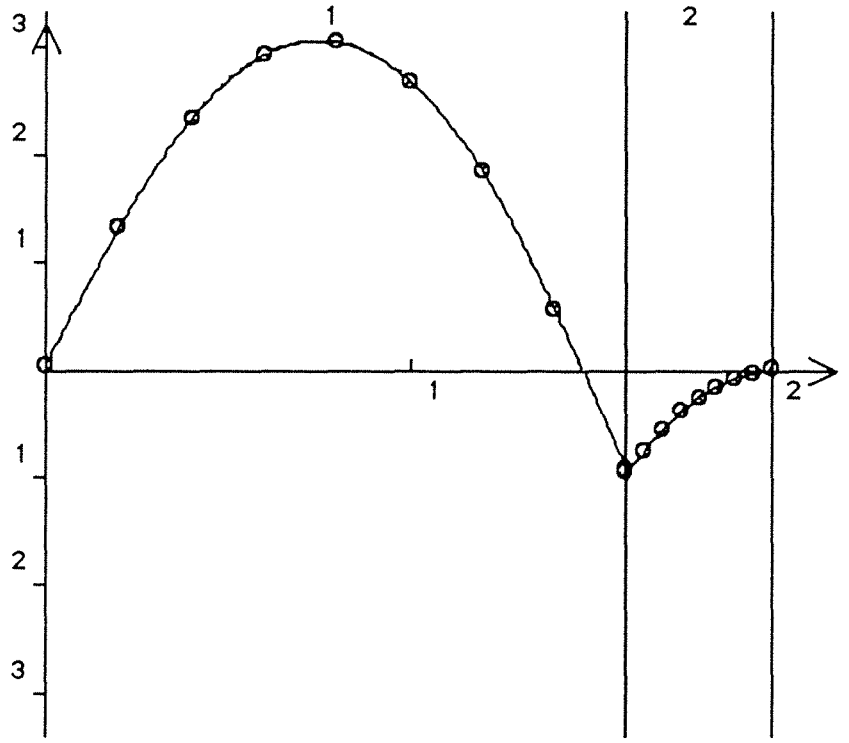


Figure VII-22-c :Tracé du moment fléchissant (le maximum : 3.05 10^{4}- N.mm)

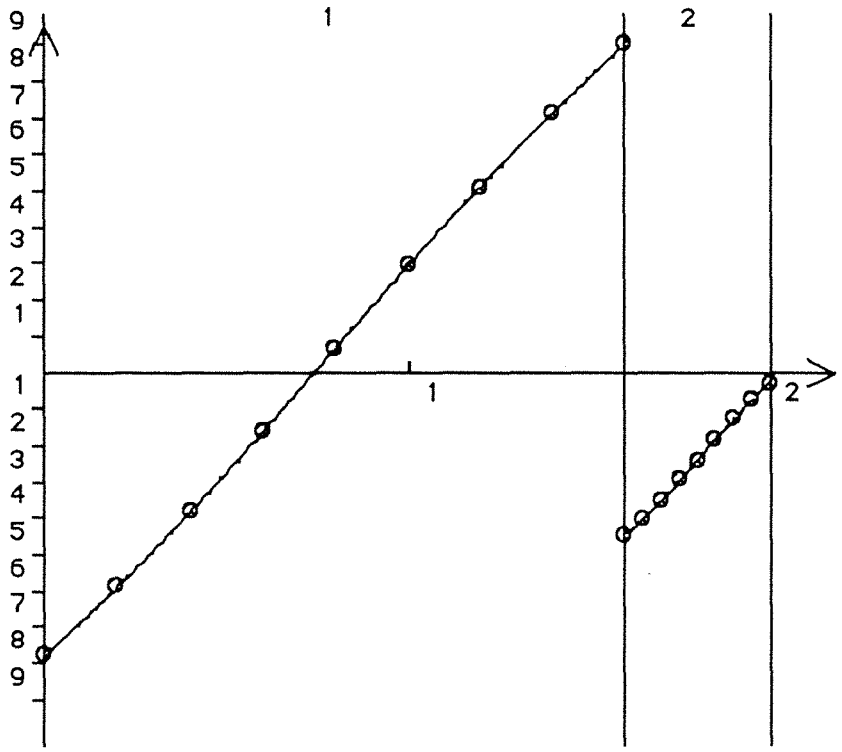


Figure VII-22-d :Tracé de l'effort tranchant (le maximum : 900 N)

VII-8-3 Croix de Saint-André

Nous nous proposons d'étudier le problème d'une croix de Saint-André (figure VII-23). La grille de départ est donnée sur la figure VII-23 : elle se compose de deux éléments de Bernoulli sur chaque montant sauf les montants diagonaux qui sont divisés en trois.

Chaque montant est constitué d'éléments de poutres identique en acier de section carrée de 10 mm de côté.

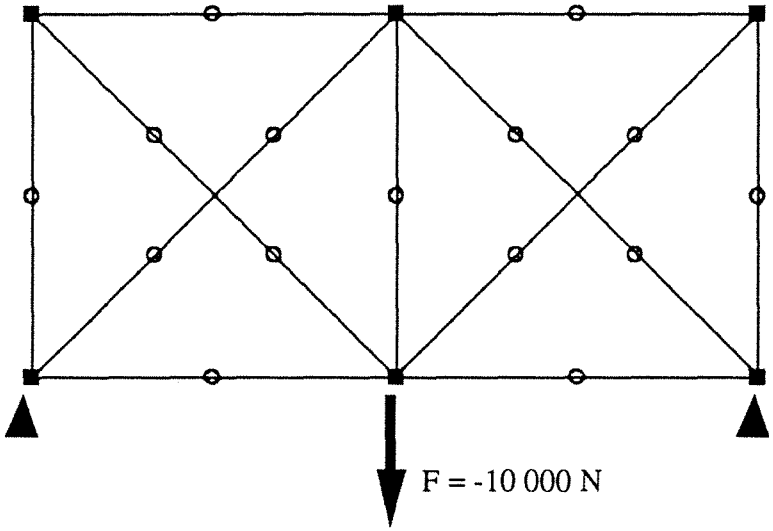


Figure VII-23 : Problème posé et grille grossière de départ.

Nous allons prendre pour référence dans ce problème, les valeurs obtenues par une méthode de résolution classique (Gradient conjugué avec pré-conditionnemment SSOR) appliquée au maillage suivant : tous les montants divisés en huit éléments sauf les montants diagonaux qui sont divisés en 10 (total 273 DDL). Tous les éléments utilisés sont des éléments de Bernoulli.

Méthode	Temps de résolution (Mac II)	résultats (valeurs maximales)
Gradient conjugué	12 minutes	u max = 0,1302 mm v max = 0,6286 mm θ max = 8,3928 10e-4 rd
Méthode multigrilles *	6 minutes	u max = 0,1302 mm v max = 0,6286 mm θ max = 8,1192 10e-4 rd

* : Nous utilisons une méthode Full multigrilles à deux grilles avec quatre itérations du processus multigrilles à chaque niveau. La convergence est extrêmement rapide dans ce cas.

Nous exploitons les diagrammes de résistance des matériaux sur les deux montants inférieurs (d'équation y=0)

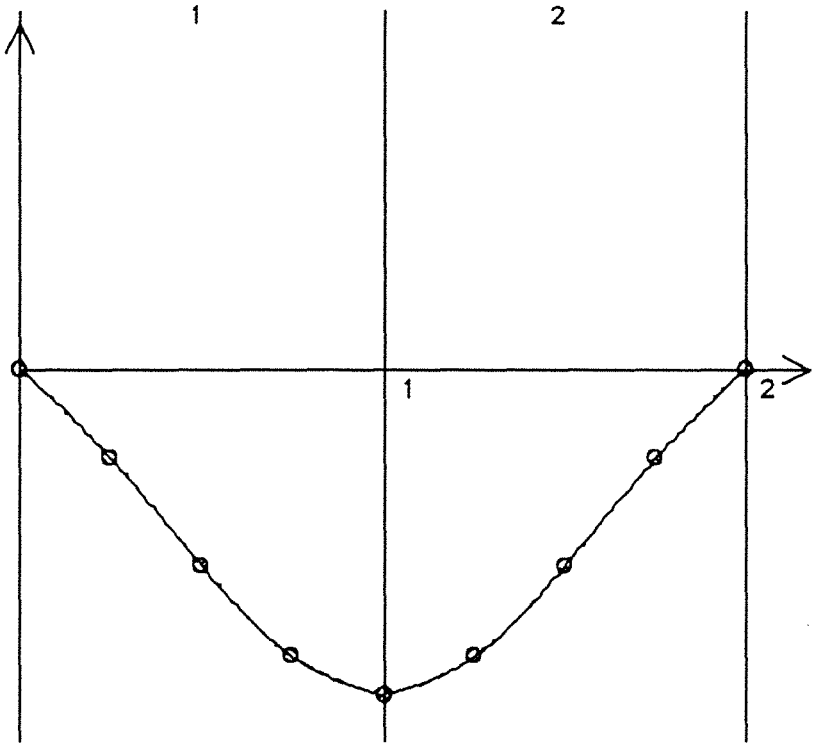


Figure VII-24-a :Tracé du déplacement v selon Oy (le maximum : 0,6286 mm)

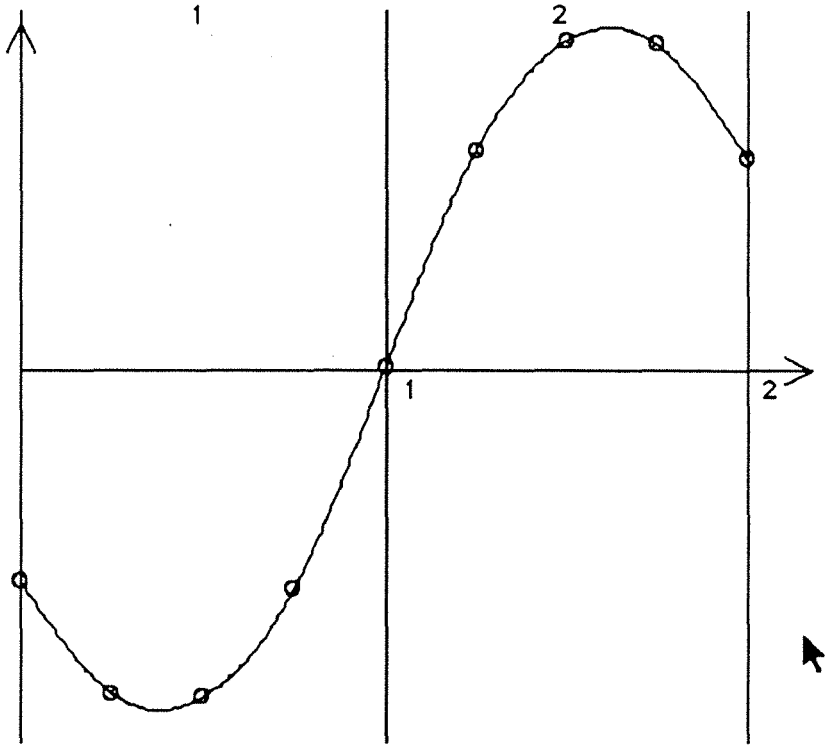


Figure VII-24-b :Tracé de la rotation θ selon Oz (le maximum : 0.812 10^{-4}rd)

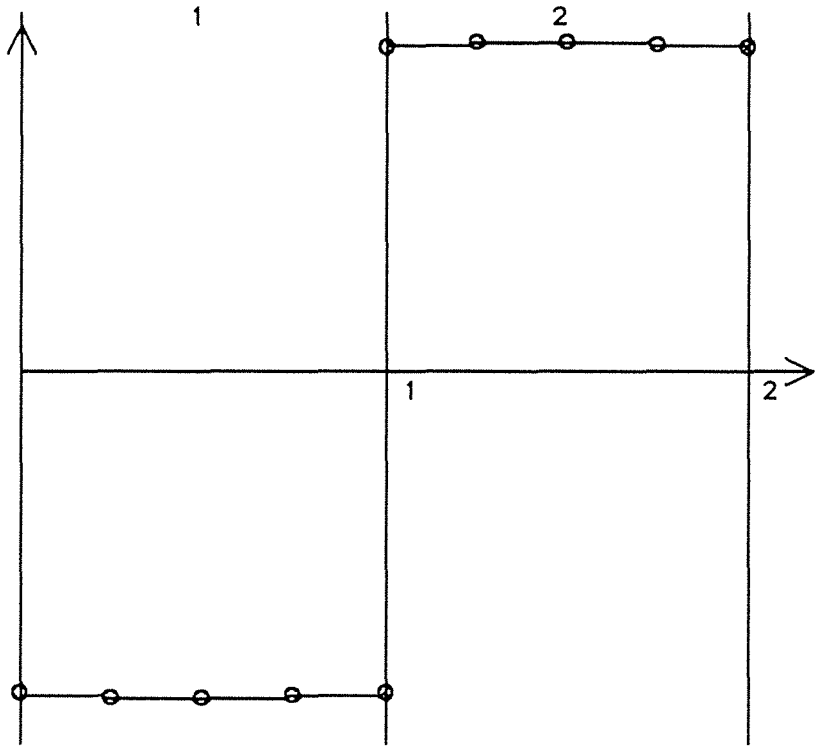


Figure VII-24-c :Tracé de l'effort tranchant T (le maximum : 7790 N)

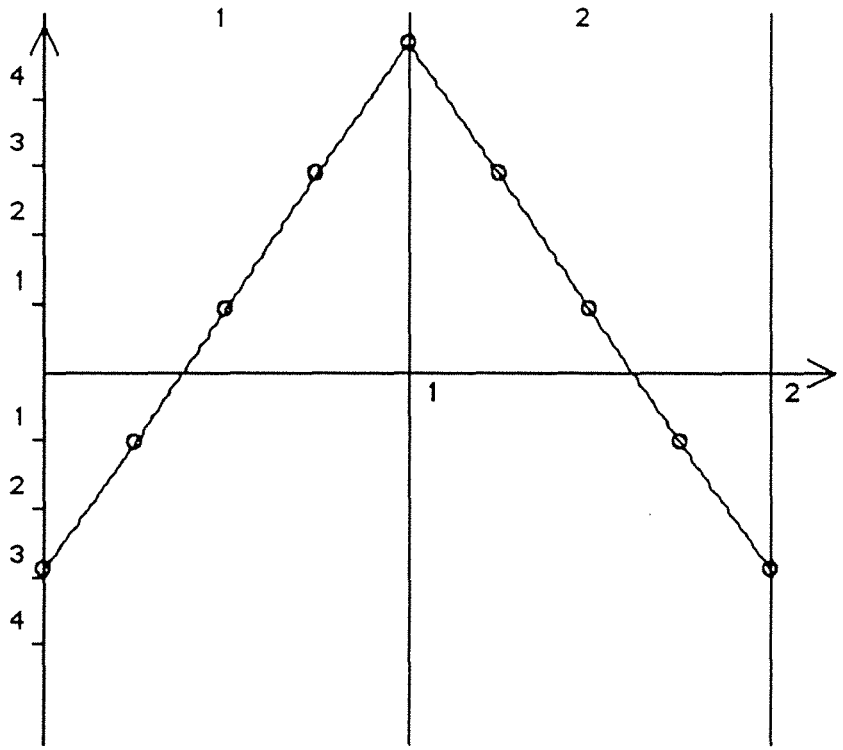


Figure VII-24-d :Tracé du moment fléchissant M (le maximum : 480 Nmm)

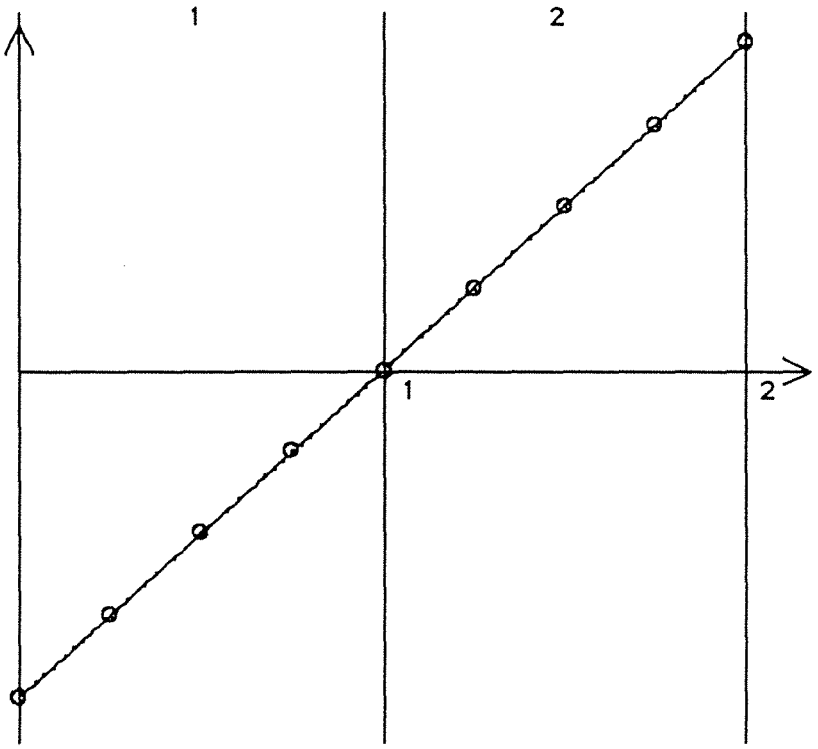


Figure VII-24-e :Tracé du déplacement u selon Ox (le maximum : 0,1080 mm)

Conclusion

Nous avons présenté ici un logiciel comptant environ 7500 instructions. Nous nous sommes efforcés de le rendre les plus convivial et le plus simple possible. Pour cela , nous avons utilisé au maximum les saisies directes à l'écran gérée par des menus déroulant. La gestion du programme par menus déroulants permet un contrôle facile des tâches à effectuer ou déjà effectuées. Cette facilité confère au logiciel (du moins nous l'espérons) la fluidité d'emploi habituelle des logiciels sur MAC INTOSH.

Ce travail de présentation est délicat, pénible et assez ingrat. Il n'en demeure pas moins que cela constitue une part importante et indispensable du travail de développement.

Parlons maintenant de l'efficacité de la méthode. La méthode multigrilles donne entièrement satisfaction dans le cadre de l'étude des ossatures de poutres planes. Cependant, comme c'était prévisible, l'introduction de la méthode multigrilles alourdit considérablement le programme et donc pénalise les temps de calcul. On peut donc se demander si l'utilisation de la méthode en vaut toujours la peine. Essayons de donner quelques éléments de réponse.

En ce qui concerne les poutres de Timoshenko, la méthode multigrilles apporte visiblement une amélioration de la solution. Par contre, cela est moins évident pour les poutres de Bernoulli. En effet ces types de poutres donnent dans le cas général des solutions tout à fait acceptables pour des discrétisations de taille moyenne. Dans ce cas, l'adjonction d'une ou plusieurs grilles supplémentaires et l'utilisation des méthodes multigrilles ne s'impose pas toujours (faible gain de précision pour un fort accroissement du temps de calcul). Il est cependant à noter que lorsque la taille du problème augmente, ces inconvénients tendent à diminuer.

Notre conclusion sera donc la suivante :

On retrouve dans l'étude des ossatures planes de poutres par les méthodes multigrilles les avantages et les inconvénients inhérents à cette méthode. Au vue de nos remarques précédentes, il ne semble pas que la méthode multigrilles trouve dans ce cas particulier son meilleur emploi, les particularités numériques et géométriques des poutres imposant certaines contraintes. La méthode multigrilles se justifie certainement plus en 2D ou en 3D surtout si l'on peut utiliser un diviseur de grille sélectif , comme nous allons le voir dans le chapitre suivant.

Chapitre VIII

Elasticité plane linéaire

VIII-1 : Introduction

Le chapitre VII trouve ici son complément par l'étude des problèmes mécaniques bidimensionnels. Nos travaux concerneront ici l'élasticité plane linéaire ; ils peuvent aussi s'appliquer à la résolution itérative de probléme d'élasticité non-linéaire. Nous allons présenter de nouveaux types de divisions de maillage conduisant (nous l'espérons) à la meilleure façon d'utiliser les méthodes multigrilles : la méthode Full-multigrilles avec maillage autoadaptatif au problème posé.

L'ensemble du logiciel répond toujours aux exigences que nous nous sommes imposées : faire un programme conversationnel et simple à utiliser utilisant au mieux les possibilités du MAC INTOSH. Pour des raisons de facilité d'emploi et de plus grande indépendance, nous avons decouplé les fonctions de maillage des fonctions de résolution. Le maillage de la pièce bidimensionnelle peut être effectué par n'importe lequel des logiciels vus aux chapitres II et III, la structure de données étant toujours compatible. Le logiciel présenté ici permet, pour un maillage donné, l'affectation des caractéristique mécaniques des matériaux, des forces, des conditions limites, puis la résolution du problème, enfin, le post-traitement de la solution. Cela représente un ensemble de 12 000 instructions.

Notre souci est aussi de présenter un logiciel aussi efficace que possible. Dans cet esprit, nous présentons un processus de raffinement autoadaptatif permettant d'affiner le maillage là où c'est nécessaire. Nous présenterons d'abord nos critères de division, puis les solutions géométriques trouvées et enfin , leurs conséquences sur notre structure de données.

VIII-2 : Elasticité plane

VIII-2-1 : Hypothèses générales de travail

• Tous les problèmes traités sont plans ou peuvent être représentés par un modèle plan (nous en verrons quelques exemples plus loin) . Nous repérons notre problème par un repère fixe $(0,\vec{x},\vec{y})$.
A chaque nœud, sont associés deux degrés de liberté globaux : les deux déplacements que nous noterons u et v.

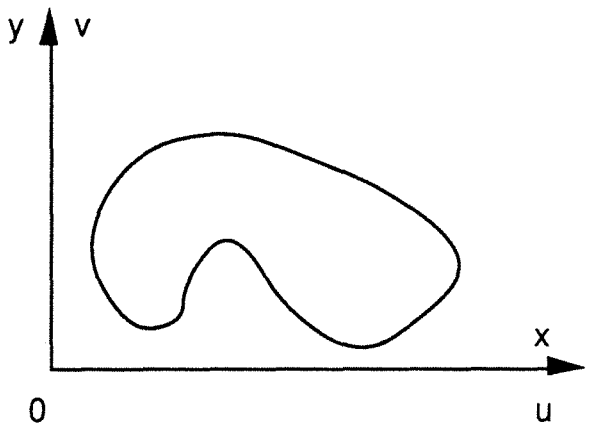


Figure VIII-1 : Représentation générale

• Nous ferons l'hypothèse des petites déformations. Nous pourrons alors confondre à tout instant $\Gamma(0)$ et $\Gamma(t)$. Nous utliserons le tenseur des contraintes de Cauchy-Euler.

• nous supposerons les matériaux utlisés homogènes isotropes et isothermes.

VIII-2-2 : Hypothèses particulières de travail

L'élasticité plane couvre trois types de problèmes : les problèmes de contraintes planes, de déformations planes et les problèmes axisymétriques. Passons les en revue.

VIII-2-2-a : Contraintes planes

Il s'agit de problèmes pour lesquels la contrainte $\sigma_{zz} = 0$. Nous supposerons le problème plan, nous aurons alors $\sigma_{xz} = \sigma_{yz} = 0$. Cela couvre les problèmes de membranes chargées dans leur plan. Les contraintes sont supposées constantes suivant l'épaisseur (supposée faible).

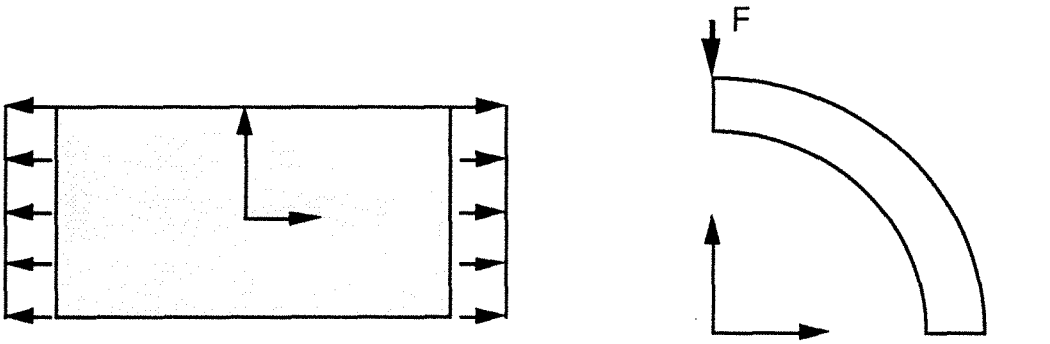


Figure VIII-2 : Exemple de contraintes planes

- les déplacement sont u et v
- les déformations se réduisent à $\langle \varepsilon \rangle = \langle \varepsilon_{xx} , \varepsilon_{yy} , \gamma_{xy} \rangle$
- les contraintes sont $\langle \sigma \rangle = \langle \sigma_{xx} , \sigma_{yy} , \tau_{xy} \rangle$
- la loi de comportement liant σ à ε s'écrit : $\{\sigma\} = [D] . \{\varepsilon\}$ (1)

$$\text{avec } [D] = \frac{E}{1-\nu^2} . \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}$$

VIII-2-2-b : Déformations planes

Cette appelation recouvre les problèmes tridimensionnels pour lesquels on suppose que u=u(x,y) , v=v(x,y) , w=0.

Cela implique que $\varepsilon_{zz} = \varepsilon_{zx} = \varepsilon_{zy} = 0$ et donc que les seules déformations non-nulles sont :

$$\langle \varepsilon \rangle = \langle \varepsilon_{xx} , \varepsilon_{yy} , \gamma_{xy} \rangle$$

Ceci peut être le cas lorsque l'on considére que l'ensemble des caractéristiques du problème (conditions limites, forces, matériaux ...etc) sont indépendant de z (par exemple, des problèmes infiniment longs).

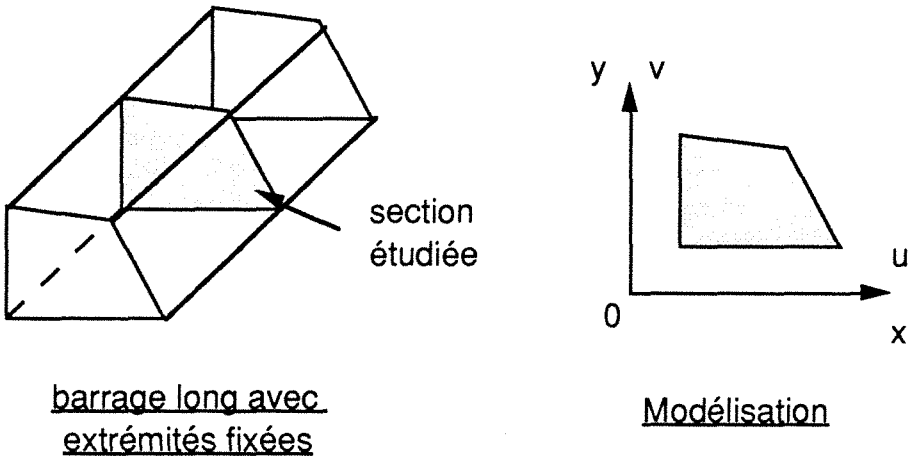


Figure VIII-3 : Exemple de déformations planes

- les contraintes sont $\langle \sigma \rangle = \langle \sigma_{xx} , \sigma_{yy} , \tau_{xy} \rangle$
- la loi de comportement liant σ à ε s'écrit : $\{\sigma\} = [D] . \{\varepsilon\}$ (2)

$$\text{avec } [D] = \frac{E}{(1+\nu) . (1-2.\nu)} . \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2.\nu}{2} \end{bmatrix}$$

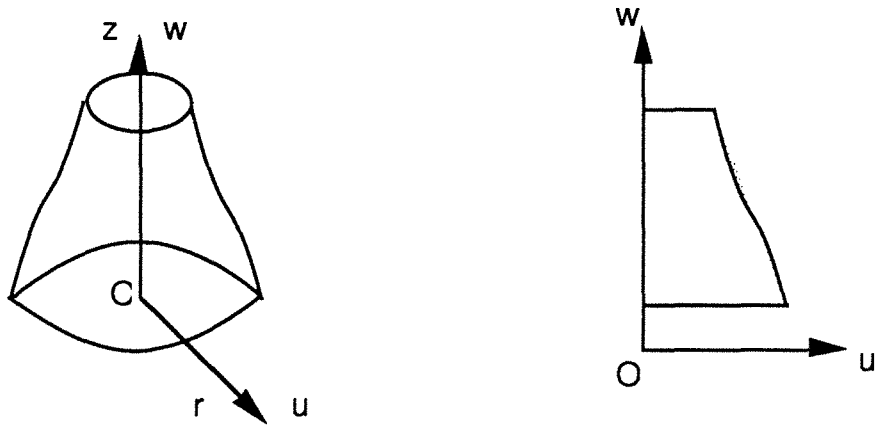
• on prend généralement de plus $\sigma_{zz} = \nu \cdot (\sigma_{xx} + \sigma_{yy})$

Les autres contraintes sont nulles.

VIII-2-2-c : axisymétrie

c"est le cas de problèmes tridimensionnels pour lesquels toutes les caractéristiques (matériaux, forces, conditions limites, etc) présentent une symétrie de révolution autour d'un axe que nous désignerons comme l'axe Oz.

Si nous construisons un repère cylindrique $(0, \vec{u}_r, \vec{u}_\theta, \vec{u}_z)$, nous constatons alors que les déplacements sont indépendants de θ. L'étude du problème peut donc se ramener à une "tranche". Nous avons alors deux degrés de liberté u (= u_r) et w (= u_z).



Problème physique Modélisation

Figure VII-4 : Exemple de problème axisymétrique

• les seules déformations non-nulles sont $\langle \varepsilon \rangle = \langle \varepsilon_{rr} , \varepsilon_{\theta\theta} , \varepsilon_{zz} , \varepsilon_{rz} \rangle$

avec $\varepsilon_{rr} = \frac{\partial u}{\partial r}$; $\varepsilon_{\theta\theta} = \frac{u}{r}$; $\varepsilon_{zz} = \frac{\partial w}{\partial z}$; $\gamma_{rz} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial r}$

• la loi de comportement lie $\{\sigma\}$ à $\{\varepsilon\}$ par $\{\sigma\} = [D] . \{\varepsilon\}$ (3)

avec :

$$\langle \sigma \rangle = \langle \sigma_{rr} , \sigma_{\theta\theta} , \sigma_{zz} , \tau_{rz} \rangle$$

et $$[D] = \frac{E}{(1+\nu) . (1-2\nu)} . \begin{bmatrix} 1-\nu & \nu & 0 & 0 \\ \nu & 1-\nu & \nu & 0 \\ 0 & \nu & 1-\nu & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}$$

VII-3 : Modélisation par éléments finis

VIII-3-1 : Théorème des travaux virtuels

Considérons un corps solide en équilibre soumis à des forces extérieures et à des déplacements imposés. Nous recherchons le champ de déplacements (approche du problème en déplacements).

Le théorème des travaux virtuels exprime le bilan des travaux virtuels internes et externes lorsque le corps est en équilibre.

$$\int_{\Omega} \sigma_{ij} . \delta\varepsilon_{ij} . d\Omega = \int_{\Omega} f_i^s . \delta u_i . d\Omega + \int_{\delta\Omega} f_i^l . \delta u_i^l . d\Gamma + \sum_i F_i . \delta u_i \qquad (4)$$

où :

Ω est le domaine considéré

$\delta\Omega$ est la frontière du domaine

f_i^s sont les forces surfaciques

f_i^l sont les forces linéiques

F_i sont les forces concentrées

σ_{ij} représente le tenseur des contraintes de Cauchy-Euler

ε_{ij} représente le tenseur des déformations infinitésimales

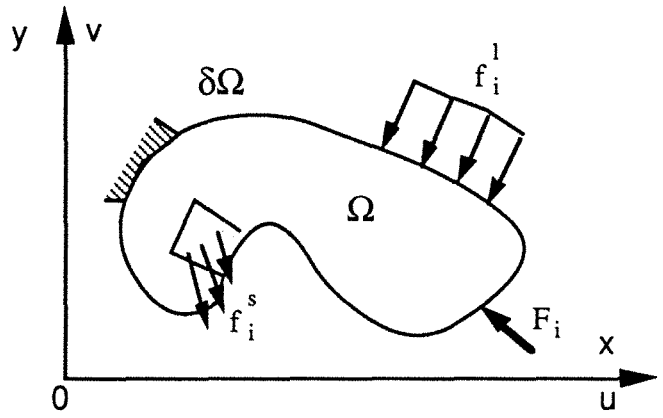


Figure VII-5a : Problème mécanique

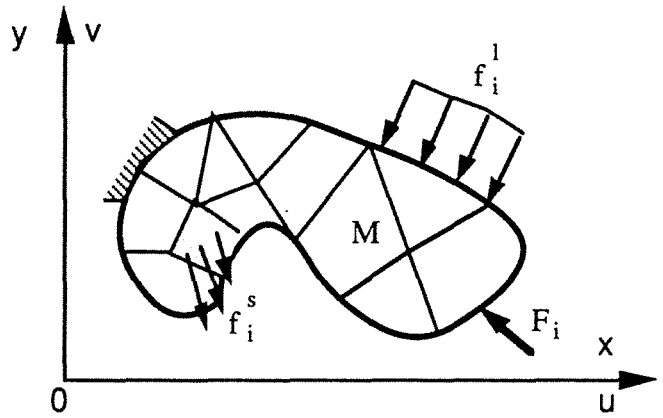


Figure VII-5b : Modélisation par éléments finis

Le déplacement d'un point M quelconque d'un élément s'exprime en fonction des valeurs aux nœuds par :

$$\begin{cases} u = \sum_{i=1}^{nn} N_i \, . \, u_i \\ v = \sum_{i=1}^{nn} N_i \, . \, v_i \end{cases} \qquad (5\,a)$$

où : nn est le nombre de nœuds par élément

u_i et v_i sont les déplacement nodaux

N_i sont les fonctions de forme de l'élément

Ce qui peut s'écrire : $\{u\} = [N] \, . \, \{\hat{u}\}$ (5 b)

avec $\{u\} = \langle u,v \rangle^t$

$$\langle \hat{u} \rangle = \langle u_1 \, , \, v_1 \, , \ldots , u_{nn} \, , \, v_{nn} \rangle$$

$$[N] = \begin{bmatrix} N_1 & 0 & \ldots & N_{nn} & 0 \\ 0 & N_1 & \ldots & 0 & N_{nn} \end{bmatrix}$$

écrivons maintenant les déformations infinitésimales :

Nous avons:

$$\{\varepsilon\} = [B] \, . \, \{\hat{u}\} \qquad (6)$$

[B] étant constitué des dérivées partielles des fonctions de forme N_i.

La loi de comportement permet d'obtenir $\{\sigma\}$ en fonction de $\{\varepsilon\}$. Elle s'écrit :

$$\{\sigma\} = [D] \, . \, \{\varepsilon\} \qquad (7)$$

Compte tenu des relations (4) à (7), nous en tirons :

$$\langle \partial u \rangle \, . \int_{\Omega} [B]^t \, . \, [D] \, . \, [B] \, . \, d\Omega = \langle \partial u \rangle \, . \int_{\Omega} [N]^t \, . \, \{f^s\} \, . \, d\Omega +$$

$$\langle \partial u \rangle \, . \int_{\delta\Omega} [N]^t \, . \, \{f^l\} \, . \, d\Gamma + \langle \partial u \rangle \, . \, \{F\} \qquad (8)$$

L'équation (8) peut alors s'écrire :

$$\langle \partial u \rangle . [K] . \{\hat{u}\} = \langle \partial u \rangle . \{f\} \quad \text{avec} \quad [K] = \int_{\Omega} [B]^t . [D] . [B] . d\Omega \qquad (9)$$

Ceci étant valable pour toute varaition virtuelle <∂u> cinématiquement admissible, nous en déduisons que l'on doit avoir :

$$[K] . \{\hat{u}\} = \{f\} \qquad (10)$$

Cette approche est tout à fait équivalente à celle consistant à dire que la première variation de l'énergie totale est nulle.

VIII-3-2 : Applications dans le cas de l'élasticité plane

Nous allons maintenant expliciter dans chaque cas la forme précise de la matrice [B] compte tenu des hypothèses du paragraphe VIII-2. Nous noterons que :

- [B] = < [B1],...,[Bn] > , les [Bi] étant détaillés par la suite
- Selon le cas, dΩ n'a pas la même forme

Contraintes planes

$$[B_i] = \begin{bmatrix} \frac{dN_i}{dx} & 0 \\ 0 & \frac{dN_i}{dy} \\ \frac{dN_i}{dy} & \frac{dN_i}{dx} \end{bmatrix} \qquad d\Omega = h.dx.dy \quad \text{où h est l'épaisseur} \qquad (11a)$$

déformations planes

$$[B_i] = \begin{bmatrix} \frac{dN_i}{dx} & 0 \\ 0 & \frac{dN_i}{dy} \\ \frac{dN_i}{dy} & \frac{dN_i}{dx} \end{bmatrix} \qquad d\Omega = dx.dy \qquad (11b)$$

axisymétrie

$$[B_i] = \begin{bmatrix} \frac{dN_i}{dr} & 0 \\ \frac{N_i}{r} & 0 \\ 0 & \frac{dN_i}{dz} \\ \frac{dN_i}{dz} & \frac{dN_i}{dr} \end{bmatrix} \qquad d\Omega = 2.\pi.r.dr.dz \qquad (11c)$$

VIII-3-3 : Choix des éléments

Pour leur rapport efficacité-simplicité, nous avons choisi les éléments suivants (Figure VIII-6) qui seront tous employés sous leur forme isoparamétrique.

les nn fonctions de forme de chaque élément sont données en annexe 6.

Triangles 3 et 6 nœuds

Quadrilatères 4, 8 et 9 nœuds

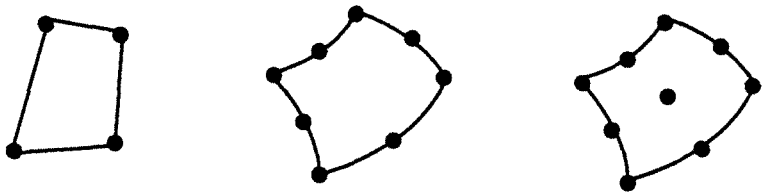


Figure VIII-6 : Les différents types d'éléments

VIII-3-4 : Intégration de la matrice élémentaire de rigidité

Le calcul de la matrice de rigidité élémentaire est effectué par une quadrature de Gauss-Legendre. Tous les calculs sont réalisés via l'élément de référence correspondant à l'élément choisi.

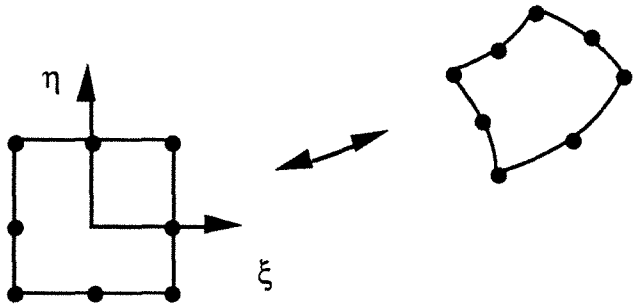


Figure VIII-7 :élément réel et élément de référence (élément 8-nœuds)

Toutes les dérivées intervenant sont transportées d'un repère à l'autre par la matrice Jacobienne (ou son inverse) :

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \eta} \end{bmatrix}$$

l'élément de surface dx.dy devient alors det(J).dx.dy.

Le nombre de point de Gauss est choisi de façon à intégrer exactement la matrice de rigidité d'un élément homothétique de l'élément de référence.

VIII-3-5 : Calcul du chargement extérieur

Nous remplaçons les forces appliquées par des forces nodales équivalentes

• Forces ponctuelles

$$\{F_{ponct}\} = \sum_{i=1}^{NFP} \{F_i\} \quad \text{où} \quad \begin{cases} \text{NFP est le nombre de forces ponctuelles} \\ \text{et} \quad \langle F_i \rangle = \langle F_{ix}, F_{iy} \rangle \end{cases} \qquad (12a)$$

• Forces surfaciques

Nous considérons un élément soumis à une accélération g faisant un angle α avec l'axe Oy. Les forces correspondantes peuvent être représentées par les forces nodales équivalentes suivantes :

$$\begin{Bmatrix} f_x \\ f_y \end{Bmatrix}_i = \int_\Omega N_i \cdot \rho.g \cdot \begin{Bmatrix} \sin(\alpha) \\ \cos(\alpha) \end{Bmatrix} \cdot d\Omega \quad \text{pour } i=1,..,nn \qquad (12b)$$

Ces forces correspondent au chargement équivalent d'un élément. Le chargement total est obtenu par assemblage (sur tous les éléments chargés surfaciquement) de tous les chargements nodaux équivalents.

• Forces linéiques

Nous considérons un élément chargé par une force $\{f_s\}$ sur une partie Γ_c de sa frontière. Nous obtenons un chargement le nodal équivalent suivant:

$$\begin{Bmatrix} f_x \\ f_y \end{Bmatrix}_i = \int_{\Gamma c} N_i \,.\, \{f_s\} \,.\, d\Gamma \quad \text{pour } i=1,..,nn \qquad (12c)$$

Il est possible de travailler à partir de forces linéiques définies de façon globale (f_{sx} et f_{sy}) ou bien de façon locale par rapport à la frontière de l'élément ($f_{s\text{-normal}}$ et $f_{s\text{-tangentiel}}$).

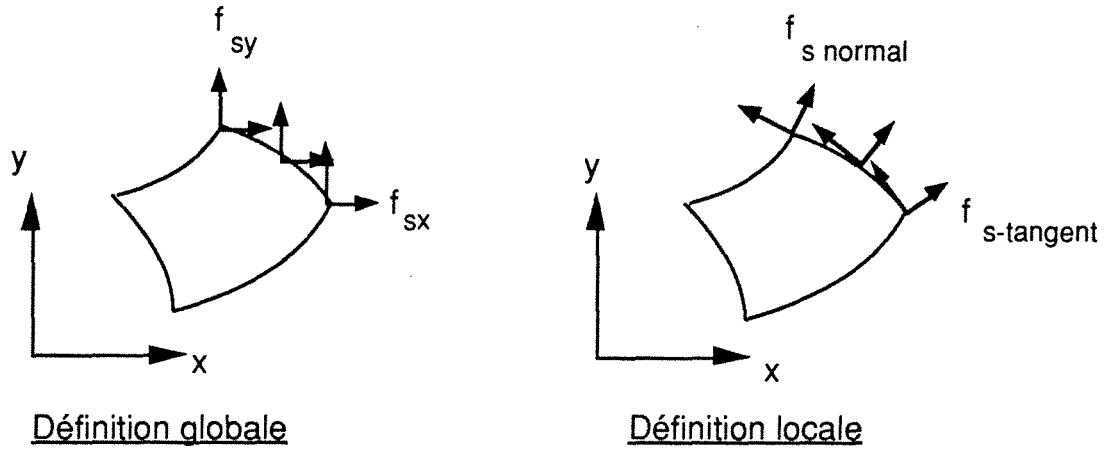


Figure VIII-8 : Exemples de cas de chargements

Toutes les intégrales intervenant sont évaluées de façon numérique par quadrature de Gauss. Les valeurs calculées sont automatiquement transférées dans le repère global.

D'un point de vue pratique, toutes les saisies de forces se font à l'écran. Les chargements crées sont stockés dans trois fichiers différents, chacun regroupant un type de forces.

Ces fichier joueront un rôle par la suite. Ils serviront, lors des opérations de divisions de grilles, à créer les nouveaux chargements correspondant. Ce point sera expliqué plus longuement par la suite.

Le chargement crée peut être vérifié ou corrigé à tous moments.

VIII-3-6 : Prise en compte des déplacements imposés

Comme dans le cas des poutres, nous pouvons imposer une valeur à un ou plusieurs degrés de liberté de certains nœuds. Le cas le plus fréquent concerne les appuis pour lesquels la on impose la valeur nulle.

Les déplacements imposés peuvent être exprimés dans le repère global ($u_x=u_{xo}$ par exemple) ou dans un repère local fixé par l'utilisateur ($u_{normal}=0$).

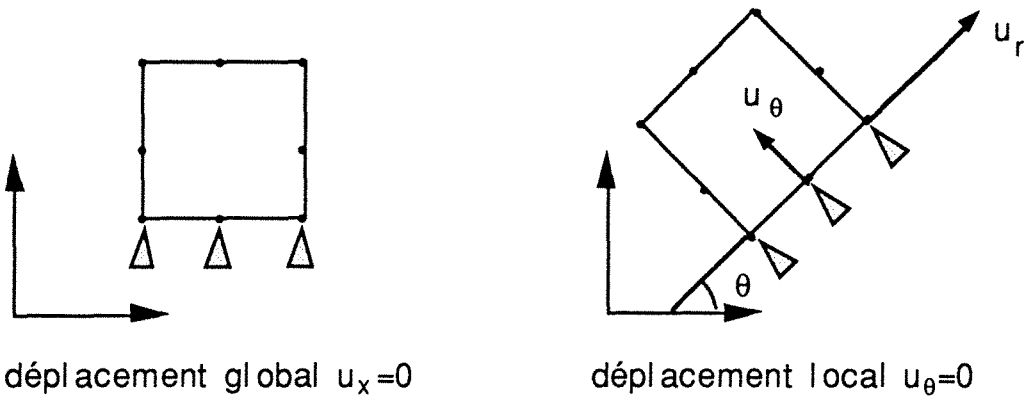


Figure VIII-9 : Exemple de déplacements imposés

Les déplacements imposés ne sont pris en compte par la méthode du coefficient dominant (vue au chapitre précédent) qu'une fois terminée l'assemblage de la matrice de rigidité.

Les déplacements imposés sont mémorisés dans un fichier disque à accès direct. Ce fichier évolue en taille au cours des créations successives de grilles lors du processus multigrilles.

VIII-3-7 : Prise en compte des matériaux

Les différents matériaux intervenant sont également définis par saisie directe à l'écran. Chaque matériau est défini par ses caractéristiques suivantes :

_ Le module d'Young : E

_ Le coefficient de Poisson : ν

_ La masse volumique : ρ

Ces informations sont stockées sur fichier disque à accès direct.

VIII-3-8 : Assemblage de la matrice de rigidité

La matrice globale est assemblée à partir des termes des matrices élémentaires de rigidité dispatchés judicieusement, de façon analogue à ce qui a été expliqué au chapitre précédent.

De même, le second membre est constitué par l'assemblage des forces appliquées.

VIII-3-9 : Algorithme général d'entrée des données

Nous pouvons résumer la structure générale de l'entrée des données par le schéma suivant (figure VII-10).

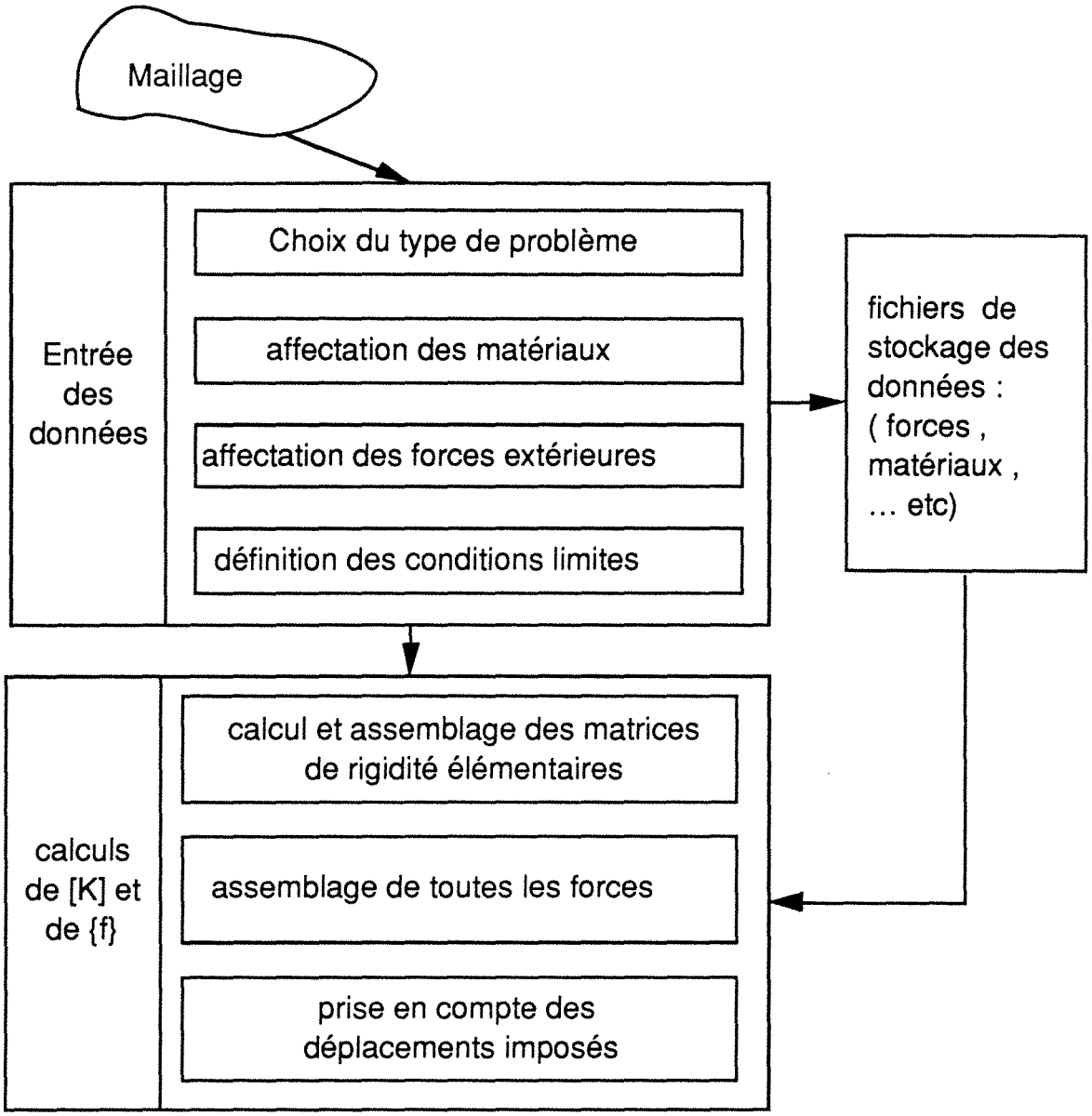


Figure VIII-10 : Algorithme général d'entrée des données

le maillage fourni doit avoir une structure conforme à celle définie au chapitre I. Cette structure comprend, comme nous l'avons vu, les données suivantes :

_ Les points par leurs coordonnées

_ Les éléments par leurs connections

_ Les arêtes et les côtés frontières

VIII-4 : Post-traitement des résultats

VIII-4-1: Visualisation de la déformée

La déformée de la structure est visualisée superposée ou non avec le maillage déformé. Elle peut être amplifiée d'un certain coefficient choisi par l'utilisateur de façon à être plus visible. Elle peut aussi être zoomée par la délimitation d'une zone à l'écran pour en discerner les détails.

VIII-4-2: Calcul des contraintes

Les éléments utilisés étant tous de classe C(0) et utilisant une approche simple en déplacement, le calcul des contraintes va poser quelques problèmes , comme nous allons le voir. Il s'agit pourtant là d'un point très important dont les résultats sont très utilisés, il convient donc de nous y appesantir.

Le calcul exact des contraintes (par dérivation des déplacements pour obtenir les déformations puis par passage aux contraintes par la loi de comportement) est effectué aux points de Gauss utilisés pour construire la matrice de rigidité. Il est prouvé que c'est en ces points que le calcul se rapproche le plus de la vraie valeur [EL 1].

Cependant, on ne peut se limiter à cela pour les deux raisons suivantes :

_ Par définition d'une modélisation par éléments finis (ou tout du moins pour ce modèle-ci) , il est plus utile d'avoir les valeurs aux nœuds qu'aux points de Gauss.

_ Le calcul des contraintes aux points de Gauss , tels que nous avons choisi ces derniers, ne donne aucune valeur sur les bords du domaine.

Malheureusement, le calcul direct des contraintes aux nœuds est impossible : les valeurs obtenues sont d'une part discontinues d'un élément à un autre et d'autre part , et surtout très différentes de leurs vraies valeurs.

Donc, pour dissiper cet inconvénient et obtenir des valeurs de contraintes aux nœuds correctes et continues, nous allons extrapoler par moindres carrés les valeurs obtenues aux points de Gauss. En voici le procédé :

La résolution du système linéaire global traduisant la discrétisation de notre problème mécanique sur une grille Ω_h nous permet d'obtenir $\{u_h\}$. Par dérivation, nous pouvons calculer les contraintes aux points de Gauss, valeurs que nous supposerons exactes. Soit σ_h cette contrainte exacte. Nous cherchons alors à obtenir les valeurs de la contrainte aux nœuds que nous noterons $\{\overline{\sigma}\} = \{\overline{\sigma}_1, .., \overline{\sigma}_{nn}\}$ où nn représente toujours le nombre de nœuds par éléments.

Nous pouvons alors écrire en utilisant les propriétés d'approximation nodale un modèle pour σ par l'équation suivante :

$$\overline{\sigma} = \sum_{i=1}^{nn} N_i . \overline{\sigma}_i \quad \text{soit encore} \quad \overline{\sigma} = \langle N \rangle . \{\overline{\sigma}\} \qquad (13)$$

Nous allons alors comparer ce modèle à σ en utilisant le principe des moindres carrés basé sur les valeurs aux points de Gauss et utilisant les $\overline{\sigma}_i$ comme paramètres. On obtient alors le système suivant:

$$\left(\int_\Omega \{N\} . \langle N \rangle . d\Omega \right) . \{\overline{\sigma}\} = \int_\Omega \{N\} . \sigma . d\Omega \qquad (14a)$$

Soit à résoudre un système du type :

$$[M] . \{\overline{\sigma}\} = \{b\} \qquad (14b)$$

avec :

$$[M] = \int_\Omega \{N\} . \langle N \rangle . d\Omega \quad \text{(matrice de masse)}$$

$$\{b\} = \int_\Omega \{N\} . \sigma . d\Omega$$

Bien évidemment, le second membre {b} (ainsi que [M] d'ailleurs) est calculé par intégration de Gauss et permet d'utiliser la valeur de la contrainte en ces points.

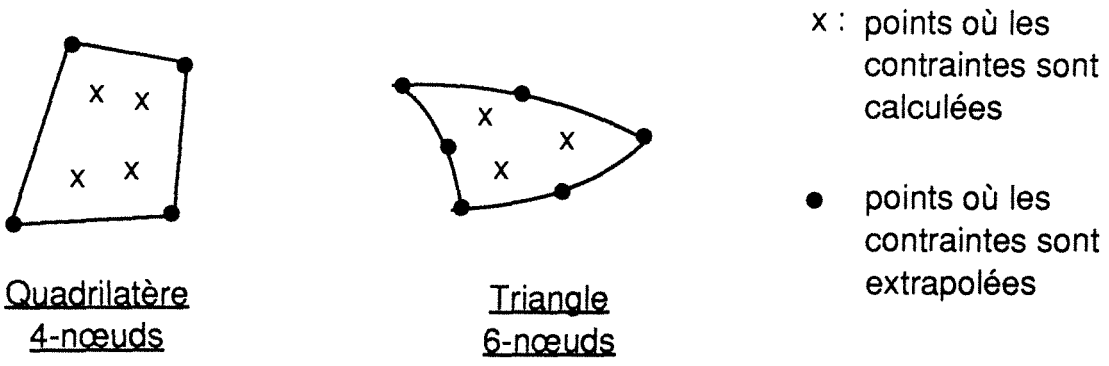


Figure VIII-11 : Exemple d'extrapolation

Nous pouvons noter que l'équation (14) a été écrite pour n'importe quelle contrainte σ. Donc , s'il y a p contraintes (p=3 ou p=4) à extrapoler (par exemple σ_{xx}, σ_{yy}, σ_{xy}), nous pouvons écrire p fois l'équation (14). La matrice [M] sera toujours la même, seul le second membre changera.

L'extrapolation exprimée par l'équation (14) peut être envisagée de deux façons :

_ Elément par élément, l'équation (14) est résolue indépendamment. Nous obtenons une "extrapolation locale" (local smoothing) [EL 3]. Elle présente l'avantage d'offrir de meilleures valeurs aux nœuds, mais cependant discontinues d'un élément à un autre.

_ En une fois sur tout le domaine. A partir de l'équation (14), on assemble globalement une matrice d'extrapolation sur tout le domaine. La résolution du système correspondant donne les contraintes aux nœuds (valeurs continues d'un élément à un autre). Cela présente l'inconvénient d'avoir parfois de facheux effets de moyenne, surtout lorsque les contraintes varient beaucoup. De plus, le système à résoudre peut être de grande taille : un système d'ordre Nt (où Nt est le nombre total de nœuds du domaine) à résoudre p fois (où p est le nombre de contraintes).

Pour le tracé des isocontraintes, nous avons choisi la seconde approche. Le système correspondant est résolu par une méthode frontale [EL 13] , ce qui économise de la place mémoire. La matrice frontale est stockée sur disque dur dans un fichier à accès direct. La matrice est triangularisée une fois pour toute puis elle est appliquée aux p seconds membres distincts. Il est à remarquer que cette méthode fournit des résultats fort curieux pour les éléments quadratiques (triangles 6-nœuds, quadrilatères 8-nœuds). Nous décomposons donc ces éléments en éléments linéaires pour pouvoir assurer une extrapolation correcte.

VIII-4 -3: Tracé des isocontraintes

Les isovaleurs sont tracées grace à l'utilisation d'un critère de contrainte équivalente (Tresca ou Von Mises) calculée à partir des valeurs aux nœuds obtenues par extrapolation globale. Le tracé est effectué sur le maillage déformé. Le nombre d'isovaleurs est choisi par l'utilisateur.

VIII-4 -4: Réactions aux appuis

A partir du vecteur global des déplacements généralisés {u}, on peut calculer [K].{u} et [K].{u}-{f}. Ceci permet d'obtenir les valeurs des réactions ainsi que de vérifier l'équilibre du système.

VIII-4 -5: Tracé d'une courbe suivant un chemin

Cette option permet d'extraire le tracé d'une valeur suivant un chemin fixé dans le domaine (une coupure en quelque sorte). Ceci peut être utile pour suivre l'évolution d'une valeur suivant un tracé qui ne correspond pas aux nœuds du maillage.

Nous limiterons les types de chemin aux segments de droite et aux arcs de cercle. Le tracé de la courbe s'appuie sur un certain nombre (choisi par l'utilisateur) de points de contrôle équirépartis sur le chemin . A partir de la position de chacun de ces points dans le repère global, nous inversons l'équation d'approximation nodale par Newton-Raphson pour obtenir les coordonnées dans le repère de référence. Ces dernières nous servent à interpoler par approximation nodale le paramétre à tracer pour obtenir une valeur à affecter à chaque point de contrôle.

Une fois ce travail terminé, les résultats sont regroupés en un tableau sont lissés par fonctions splines (voir Annexe 5) et tracés à l'écran.

De façon générale, toutes les valeurs nodales (déplacements, contraintes, etc...) ou aux points de Gauss sont disponibles numériquement. Il est aussi possible de sauvegarder numériquement une étude (maillage+forces+déplacements+contraintes) en fichier disque afin de pouvoir reprendre une étude sans avoir à refaire les calculs. Cette option est disponible à tous les niveaux de grilles.

VIII-5 : Division adaptative de maillage

VIII-5-1 : Introduction

Dans le cas bidimensionnel, le cycle Full-multigrilles choisi semble particulièrement avantageux dans la mesure ou il définit les grilles de façon dynamique. Cependant, une division uniforme d'un maillage bidimensionnel n'est pas toujours avantageuse dans la mesure où elle raffine la maillage dans tout le domaine , donc en particulier là où ce n'est pas nécessaire.

Notre idée est donc d'essayer de faire une division du maillage qui "s'adapte au problème". Cette idée va s'articuler autour de deux questions-clés: d'abord sur quel critère décider de ce qui doit être divisé ? ; ensuite quelle solution géométrique trouver ?. Nous allons dans ce paragraphe, essayer d'apporter quelques solutions.

VIII-5-2 : Critères de divisions

VIII-5-2-a : Critères de division en norme de l'énergie

Nous considérons un domaine Ω sur lequel nous cherchons à résoudre un problème mécanique dont la discrétisation sur une grille Ω_h conduit au système :

$$[K_h].\{u_h\} = \{f_h\} \quad (15)$$

Nous supposerons [K] symétrique définie positive.

La résolution de ce système conduit à $\{u_h\}$ qui constitue une approximation de la solution théorique exacte du problème u_{ex}. Si nous notons $\{u\}$ le vecteur formé des déplacements exacts (construits à partir de u) aux nœuds) , nous pouvons définir une erreur sur les déplacements :

$$\{e_u\} = \{u_{ex}\} - \{u_h\} \qquad (16a)$$

Nous pouvons aussi définir une erreur sur les contraintes :

$$\{e_\sigma\} = \{\sigma_{ex}\} - \{\sigma_h\} \qquad (16b)$$

Il est alors naturel de définir une norme de l'énergie par :

$$\|v\|^2 = \int_{\Omega_h} \langle v \rangle . [K_h] . \{v\} . d\Omega \qquad (17)$$

Si l'on applique cette norme à l'erreur sur les déplacements, on obtient :

$$\|e_u\|^2 = \int_{\Omega_h} \langle u_{ex} - u_h \rangle . [K_h] . \{u_{ex} - u_h\} . d\Omega \qquad (18)$$

Appliquée dans le cas de l'élasticité plane, cette dernière équation s'écrit alors :

$$\|e_u\|^2 = \int_{\Omega_h} \langle \varepsilon_{ex} - \varepsilon_h \rangle . \{\sigma_{ex} - \sigma_h\} . d\Omega \qquad (19a)$$

$$\|e_u\|^2 = \int_{\Omega_h} \langle \sigma_{ex} - \sigma_h \rangle . [D]^{-1} . \{\sigma_{ex} - \sigma_h\} . d\Omega \qquad (19b)$$

ceci définissant l'énergie de l'erreur de déformation.

Cette norme est trés utilisée car elle est représentative du problème d'élasticité plane. Cependant, d'autres normes sont possibles, la plus fréquemment utilisée étant la norme L_2 :

$$\|v\|_{L2}^2 = \int_{\Omega_h} \langle v \rangle . \{v\} . d\Omega \qquad (20)$$

Dans notre cas, nous choisirons la norme de l'énergie plus adaptée à notre problème.

Appliquons maintenant ces resultats dans le cas d'un maillage Ω_h constitué de NEL éléments. Nous pourrons alors écrire :

$$\|e_u\|^2 = \sum_{i=1}^{NEL} \|e_i\|^2 \qquad (21)$$

Cette dernière relation nous donne deux indications précieuses :

_ D'abord, la norme totale de l'erreur sur Ω

_ Ensuite, la façon dont cette erreur se répartit sur chaque élément.

L'idée de base du critère de division est de calculer sur chaque élément une erreur relative en comparant $\|e\|$ à $\|u_h\|$. Sur chaque élément i, nous imposons le fait que $\|e_i\|$ ne doit pas dépasser un certain pourcentage k de l'énergie totale moyenne ; soit:

$$\|e_i\| < k \cdot \frac{\|u_h\|}{NEL} \qquad NEL = \text{nombre d'éléments} \qquad (22a)$$

ou encore:

$$\|e_i\| < k \cdot \overline{\|u\|} \qquad \text{où} \quad \overline{\|u\|} = \frac{\|u_h\|}{NEL} = \text{énergie moyenne} \qquad (22b)$$

Zienckiewicz propose une valeur de k = 0,05 (5 %) [EL 1] comme valeur de travail.

Les éléments dont le rapport $\|e_i\| / k \cdot \overline{\|u\|}$ est inférieurs à un certain seuil (voisin de 1) sont à traiter (dans notre programme, ils seront subdivisés, mais dans un cas plus général, le traitement pourra être autre, une augmentation du nombre de nœuds par exemple).

Il reste cependant un problème majeur : si $\|u_h\|$ est aisé à calculer (le calcul est evidemment fait par intégration de Gauss pour utiliser les valeurs des contraintes en ces points), le calcul exact de $\|e_u\|$ est très difficile vu que l'on ne connait pas les contraintes exactes σ_{ex} indispensables au calcul.

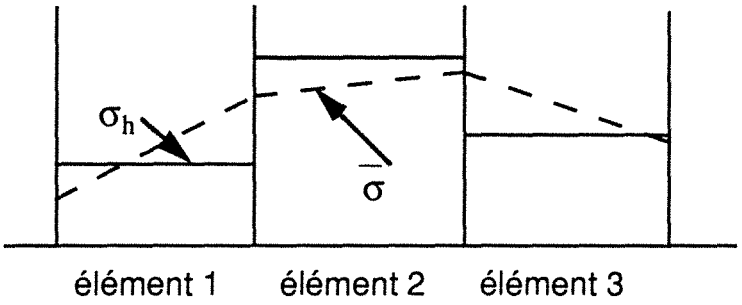
Nous allons donc utiliser une approximation de $\sigma_{ex} - \sigma_h$ proposée par Zienckiewicz [EL 1] :

Approcher $\sigma_{ex} - \sigma_h$ par $\bar{\sigma} - \sigma_h$

où $\bar{\sigma}$ sont les contraintes continues lissées par moindres carrés comme cela a été expliqué en VIII-4-2

cette valeur $\bar{\sigma} - \sigma_h$ fournit à un moindre coût une bonne approximation de $\sigma_{ex} - \sigma_h$, chose qui a été prouvé par des essais numériques [EL 1]].

Bien sûr, pour calculer la norme de l'énergie , nous utiliserons une intégration de Gauss pour pouvoir avoir les meilleures valeurs de σ_h.



exemple unidimensionnel

$\bar{\sigma}$ = valeur de la contraintee lissée par moindres carrés

σ_h = valeur discontinue obtenue à partir des points de Gauss

Figure VIII-12 : Un exemple unidimensionnel des contraintes utilisées

La mise en œuvre du critère peut utilement se résumer par l'algorithme suivant :

Résolution de $[K_h].\{u_h\} = \{f_h\}$ donnant les déplacements $\{u_h\}$

Pour tous les éléments i faire

- 1/ calcul de $[M_i] = \int_{\Omega i} \langle N \rangle . \{N\} \, . \, d\Omega$
- 2/ calcul de $\{b_i\} = \int_{\Omega i} \{N\} \, . \, \sigma_h \, . \, d\Omega$
- 3/ calcul de $\|u_h\|_i^2 = \int_{\Omega i} \langle \sigma_h \rangle \, . [D]^{-1} . \{\sigma_h\} \, . \, d\Omega$
- 4/ assemblage de $[M_i]$ et de $\{b_i\}$ dans le système :

$$[M] . \{\bar{\sigma}\} = \{b\} \quad \text{où } \{\bar{\sigma}\} \text{ est l'inconnue}$$

$$\text{et sommation } \|u_h\|^2 = \|u_h\|^2 + \|u_h\|_i^2$$

Fin de la boucle

Résolution par méthode frontale de $[M] . \{\bar{\sigma}\} = \{b\}$

pour chaque élément i faire

calculer $\|e\|_i^2 = \int_{\Omega i} \langle \bar{\sigma} - \sigma_h \rangle \, . [D]^{-1} . \{\bar{\sigma} - \sigma_h\} \, . \, d\Omega$

et comparer avec l'énergie moyenne $\overline{\|u\|} = \dfrac{\|u_h\|}{NEL}$

Fin de la boucle

Figure VIII-13 : Algorithme d'évaluation de l'erreur utilisant la norme de l'énergie

Ce critère de maillage donne de bons résultats. Il a l'avantage d'être assez facile à mettre en œuvre dans le cas de l'élasticité plane et d'être généralisable à tous les éléments utilisés par notre programme. Dans l'optique d'une utilisation dans un processus multigrilles, nous avons préféré prendre un critère de division un peut moins sévère que 5 %, nous avons donc préféré prendre un taux de 10 %.

VIII-5-2-b : Critères de division sur les équations d'équilibre

Nous reprenons les notations du paragraphe précedent (VIII-5-2-a).

Nous avons toujours :

$$\|u_{ex} - u_h\|^2 = \int_\Omega \left\langle \varepsilon_{ex} - \varepsilon_h \right\rangle . \left\{ \sigma_{ex} - \sigma_h \right\} . d\Omega \qquad (23a)$$

$$\|u_{ex} - u_h\|^2 = \int_\Omega \sum_{i,j} \left(\varepsilon_{ex\ ij} - \varepsilon_{h\ ij} \right) . \left(\sigma_{ex\ ij} - \sigma_{h\ ij} \right) . d\Omega \qquad (23b)$$

Le solide en équilibre soumis à des forces répartie f_i sur Ω, à des efforts de traction t_i sur $\delta\Omega/\Gamma_u$ et devant répondre à des déplacements imposés sur Γ_u doit vérifier les équations suivantes :

$$\begin{cases} \sigma_{ex\ ij,j} + f_i = 0 \quad \text{dans } \Omega & (24a) \\ \sigma_{ex\ ij} . n_j = t_i \quad \text{sur } \delta\Omega/\Gamma_u & (24b) \\ u = u_o \quad \text{sur } \Gamma_u & (24c) \end{cases}$$

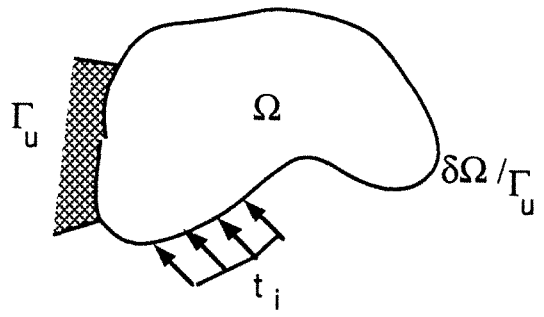


Figure VIII-14 : Définition du problème

La résolution du problème sur la grille discrète Ω_h donnent des solutions approchées pour u et σ qui font que les équations (24a) et (24b) ne sont plus exactement vérifiées. On notera cependant que l'on peut s'assurer par divers procédés que l'équation (24c) soit vérifiée. Nous obtenons alors :

$$\begin{cases} \sigma_{h\ ij,j} + f_i = r_i \quad \text{dans } \Omega_h & (25a) \\ \sigma_{h\ ij} \cdot n_j - t_i = q_i \text{ sur } \Omega_h / \Gamma_u & (25b) \\ u = u_o \text{ sur } \Gamma_u & (25c) \end{cases}$$

Revenons maintenant à la norme de l'erreur $\|u_{ex} - u_h\|$ que nous allons écrire comme la somme de ses composantes sur chaque élément Ω_i composant le maillage de Ω. En utilisant les formules de Green ainsi que les relations (24) et (25), on obtient alors :

$$\|u_{ex} - u_h\|^2 = \sum_{\Omega i} \left(\int_{\Omega i} \sum_i r_i \cdot \left(u_{ex\ i} - u_{h\ i} \right) . d\Omega + \right.$$

$$\left. \int_{\delta\Omega i} \sum_{i,j} \left(\sigma_{ex\ ij} - \sigma_{h\ ij} \right) . n_j . \left(u_{ex\ i} - u_{h\ i} \right) . d\Gamma \right) \qquad (26)$$

ce que nous écrirons $\|u_{ex} - u_h\|^2 = \sum_{\Omega i} \left(J_{1\,i} + J_{2\,i} \right)$

divisons la seconde J_2 intégrale en deux : une partie relative à la portion de Ω_i appartenant à $\delta\Omega/\Gamma_u$ (ce sont les arêtes externes, que nous noterons Ar.ext dans nos formules) et une autre relative aux arêtes internes issues de la discrétisation (que nous noterons Ar.int).

Nous obtenons alors :

$$\sum_{\Omega i} J_{2\,i} = \sum_{Ar.int} \int_{Ar.int} \sum_{i,j} \left(\sigma^{e}_{h\,ij} - \sigma^{e'}_{h\,ij} \right) . n_j^{e-e'} . \left(u_{ex\,i} - u_{h\,i} \right) . d\Gamma -$$

$$\sum_{Ar.ext} \int_{Ar.ext} \sum_{i} q_i . \left(u_{ex\,i} - u_{h\,i} \right) . d\Gamma$$

où $\sigma^{e}_{h\,ij}$ et $\sigma^{e}_{h\,ij}$ représentent les valeurs de $\sigma_{h\,ij}$ sur les éléments e et e' dont l'intersection donne l'arête interne Ar.int et $n^{e-e'}$ représente la normale à cette arête.

Nous en déduisons que la norme de l'erreur provient de trois origines différentes :

1/ Le non respect des équations d'équilibre par le tenseur des contraintes.

2/ Les discontinuités des contraintes à l'interface entre deux éléments.

3/ Le non-respect des forces imposées sur les frontières.

En utilisant pour toutes ces intégrales l'inégalité de Cauchy-Schwartz, on trouve :

$$\|u_{ex} - u_h\|^2 < \sum_{\Omega i}\left(\sum_i \int_{\Omega i} r_i^2 . d\Omega\right)^{1/2} . \|u_{ex} - u_h\|_{(L2\ \Omega i)} +$$

$$\sum_{Ar.int}\left(\sum_{i,j} \int_{Ar.int} \left(\left(\sigma^{e}_{h\,ij} - \sigma^{e'}_{h\,ij}\right) . n_j^{e-e'}\right)^2 . d\Gamma\right)^{1/2} . \|u_{ex} - u_h\|_{(L2\ Ar.int)} +$$

$$\sum_{Ar.ext}\left(\sum_i \int_{Ar.ext} q_i^2 . d\Gamma\right)^{1/2} . \|u_{ex} - u_h\|_{(L2\ Ar.ext)} \qquad (28)$$

Où $\|\ .\ \|_{(L2\,\Omega)}$ désigne la norme L_2 sur le domaine Ω.

En faisant l'hypothèse que les écarts de contraintes sont égalements distribués entre deux éléments voisins; en utilisant l'inégalité de Poincaré et la coercitivité de l'opérateur énergie de déformation ainsi que les propriétés des normes utilisées, on montre [EL 18] que l'on peut alors écrire :

$$\|u_{ex} - u_h\| < C' . \left(\sum_{\Omega i}\left(\|r\|_{(L2\ \Omega i)} + \|q\|_{(L2\ Ar.ext\Omega i)} + \|\|s\|\|_{(L2\ Ar.int\Omega i)}\right)^2\right)^{1/2} \qquad (29)$$

avec :

$$\|r\|_{(L2\ \Omega i)} = \left(\int_{\Omega i} \sum_i \left(\sum_j \sigma_{h\,ij,j} + f_i\right)^2 . d\Omega\right)^{1/2}$$

qui représente la norme du résidu hors-équilibre sur l'élément Ωi.

$$\|q\|_{(L2\ Ar.ext\Omega i)} = \left(\int_{Ar.ext\Omega i} \sum_{i} q_i^2 \, . \, d\Gamma \right)^{1/2}$$

qui représente la norme du défaut hors-équilibre sur les arêtes extérieures de l'élément Ωi.

$$|||s|||_{(L2\ Ar.int\Omega i)} = \frac{1}{2} \cdot \left(\int_{Ar.int\Omega i} \sum_{i,j} \left(\sigma_{h\,ij}^{\Omega i} - \sigma_{h\,ij}^{\Omega k} \right)^2 . \, d\Gamma \right)^{1/2}$$

qui représente la norme du saut de contraintes sur les arêtes intérieures de l'élément Ωi.

Il est montré que si les contraintes sont continues aux frontières internes des éléments (ie $|||s||| = 0$) et si les équations d'équilibre sont vérifiées (ie $\|r\|] = 0$), alors les conditions limites en contraintes sur $\delta\Omega/\Gamma_u$ où s'exercent les efforts sont satisfaites, ce qui entraine que $\|q\| = 0$ [EL 18].

Il s'ensuit donc que le premier et le troisième terme suffisent à définir un critère d'erreur.

Nous avons donc à calculer sur chaque élément Ωi une erreur sous la forme suivante :

$$\delta_{\Omega i} = \|r\|_{(L2\ \Omega i)} + |||s|||_{(L2\ Ar.int\Omega i)} \qquad (30)$$

Certains auteurs [IN 5] pensent que $\|r\|$ est souvent négligeable par rapport à $|||s|||$, et que donc celui-ci suffit à définir le critère. Cependant, certains essais numériques [EL 18] semblent montrer qu'il n'en est rien, nous gardons donc la version complète du critère.

Une autre remarque judicieuse [EL 18] précise que si l'on décide de diviser les éléments jugés trop grossiers, on a tout intérêt à pondérer les deux normes par la surface et le contour respectif de l'élément, soit de prendre :

$$\delta_{\Omega i} = \left(\int_{\Omega i} d\Omega\right)^{1/2} . \|r\|_{(L2\ \Omega i)} + \left(\int_{\delta\Omega i} d\Gamma\right)^{1/2} . \|s\|_{(L2\ Ar.int\Omega i)} \qquad (31)$$

de façon à diviser les éléments les plus gros.

Nous ne pouvons donner un sens à ce calcul d'erreur que de façon relative. Notre critère s'énnocera donc de la façon suivante :

- Sur chaque élément Ω_i , calcul de $\delta_{\Omega i}$
- Calcul de la valeur moyenne sur Ω : $\overline{\delta}$
- Sur chaque élément Ωi, calcul de $\delta_{\Omega i}/\overline{\delta}$. Les éléments dont ce rapport dépasse 1 sont à diviser.

Il reste dependant un obstacle à lever. Pour pouvoir évaluer $\delta_{\Omega i}$, il est nécessaire d'avoir une bonne valeur des contraintes aux nœuds et qui soient discontinues. Pour cela , nous allons procéder à un lissage local (vu au paragraphe VIII-4-2) sur chaque élément, ce qui nous permettra d'extrapoler les valeurs des contraintes aux nœuds.

L'équation de lissage est résolue sur chaque élément séparément. Nous rappelons qu'elle s'écrit :

$$[M] . \{\overline{\sigma}\} = \{b\} \qquad (32)$$

$$\text{avec} \begin{cases} [M] = \int_{\Omega i} \langle N \rangle . \{N\} . d\Omega \\ \{b\} = \int_{\Omega i} \{N\} . \sigma . d\Omega \end{cases}$$

Dans notre cas {N} représente les fonctions de forme pour les éléments triangulaires 3-nœuds et quadrilatères 4-nœuds. Pour les autres éléments (quadratiques) , nous décomposons préalablement en éléments linéaires sur lesquels le lissage est effectué.

Grace aux valeurs de $\overline{\sigma}$, nous pouvons évaluer à la fois $\|r\|$ et $|||s|||$. On pourra noter une méthode astucieuse [EL 18] permettant de calculer $\|r\|$ par intégration par parties sans avoir à redériver explicitement le champ des contraintes.

L'algorithme de la méthode peut se résumer de la façon suivante :

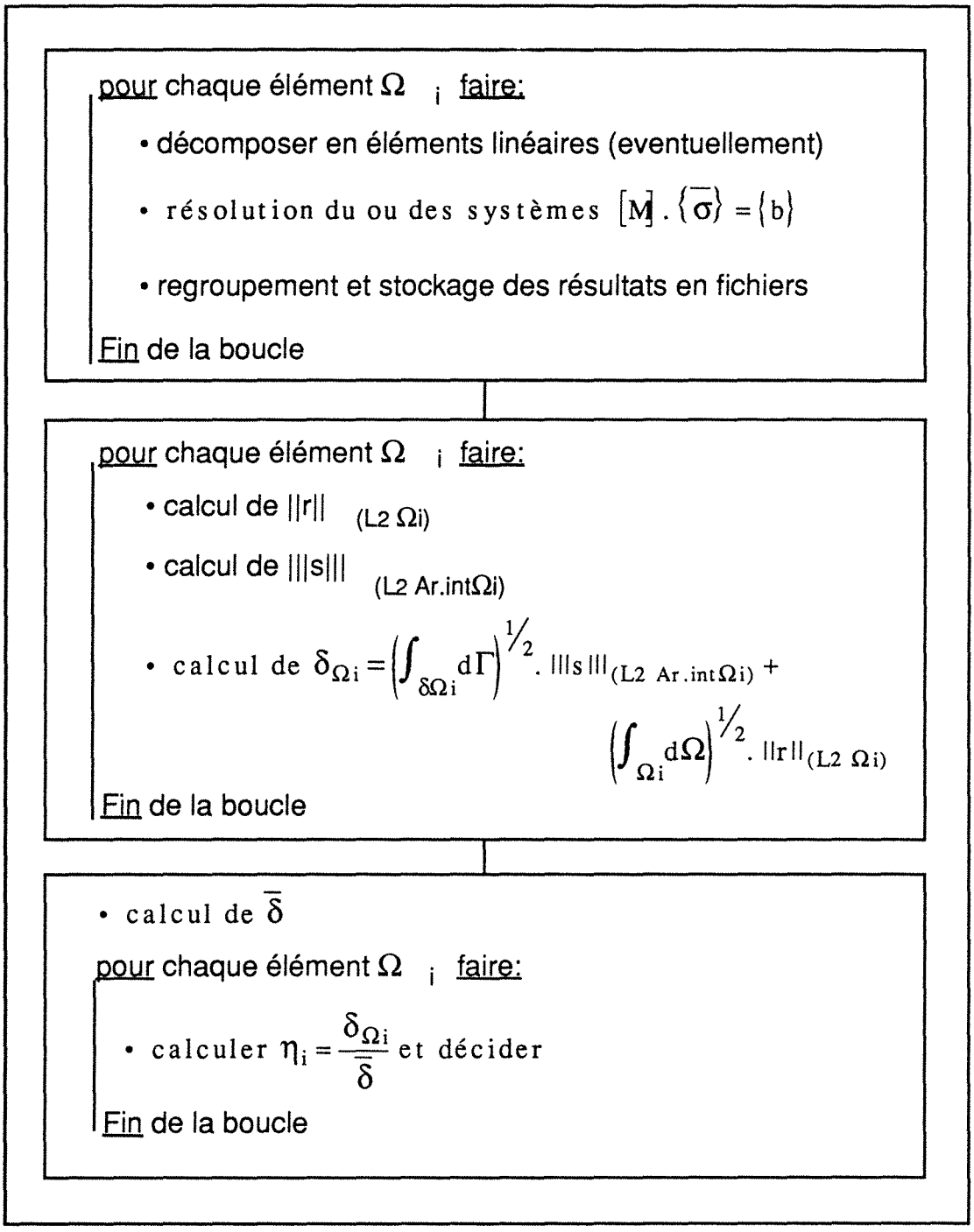


Figure VIII-16 : algorithme de calcul de l'erreur relative sur Ω

Voici présentés ces deux critères d'évaluation d'erreur permettant d'avoir une idée de la

qualité du maillage et de savoir où celui ci convient le moins bien.

On pourra remarquer que ces deux critères sont indépendants des modules de résolution à proprement parler. Ils sont utilisables comme "post-traitement".

D'un point de vue pratique, les deux critères donnent desrésultats satisfaisants, nous avons cependant décidé d'utilser le second critère qui semble meilleur et traduisant mieux le problème mécanique (heureusement, car il est tout de même plus lourd à mettre en œuvre).

VIII-5-3 : Réduction de la norme de l'erreur

Nous venons juste de de voir que l'on pouvait mettre en œuvre des critères indiquant sur le maillage la "qualité" de chaque élément. En comparant cette "qualité" avec la valeur moyenne sur tout le maillage, on est capable de savoir quels sont les éléments qui sont mal adaptés au problème posé.

Cette notion de critère porte en soi deux indications intéressantes. D'abord, nous sommes capables de savoir où est-ce que le maillage pose problème. Nous avons donc la possibilité de prévoir les zones délicates quel que soit le problème (d'élasticité plane) à résoudre. Ensuite, et lié à la remarque précédente, nous entrevoyons la possibilité de ne raffiner le maillage que là où cela est nécessaire, ce qui permet de réduire la taille du nouveau maillage en conservant le plus possible la maillage de départ.

Lorsque l'on a déterminé quels éléments sont à traiter, nous avons plusieurs possibilités. Parmi les plus usitées [EL 1] , citons :

- _ utiliser des éléments mixtes déplacements-contraintes plus puissant mais plus lourd à manipuler.
- _ utiliser des méthodes itératives destinées à améliorer la continuité du champ de contraintes en modifiant le champ de déplacement [EL 6].
- _ deformer localement le maillage pour ramener des nœuds vers les singularités (r-raffinement).
- _ diviser les éléments à traiter en éléments de même type mais de taille plus petite (h-raffinement).
- _ Garder le maillage de départ, mais augmenter le degrés des fonctions d'interpolation des éléments incriminés (p-raffinement).

Nous pouvons résumer cette méthodologie par la figure suivante (figure VIII-17)

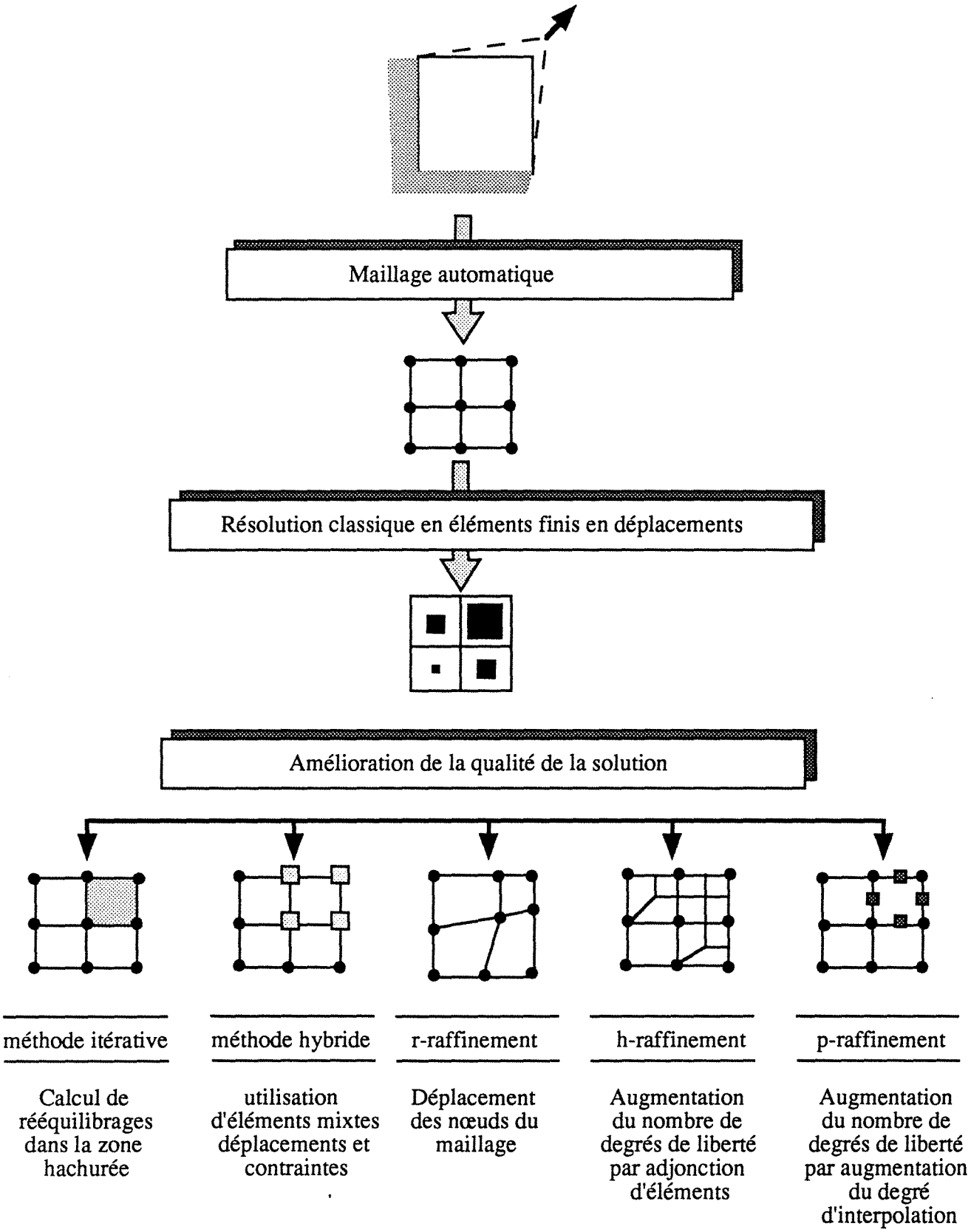


Figure VIII-17 : divers types de raffinement
(d'après un document du CETIM)

Nous avons choisi la première solution (h-raffinement) pour résoudre notre problème. Notre processus se divise en trois étapes :

1_ Evaluation du critère d'erreur et définition des éléments à diviser

2_ Division des éléments à traiter

3_ Raccord entre le reste du maillage et les éléments divisés de façon à ce que le résultat soit compatible avec la notion d'éléments finis.

Revenons en détail sur ces étapes. Nous avons déjà été vue. Pour la deuxième, nous divisons chaque élément à traiter de façon homogéne en 4 ou en 9 pour les éléments triangulaires, en 9 pour les éléments quadrangulaires (pour des raisons de maillage, une division en 4 n'a pas de solution géométrique) selon les schémas suivants :

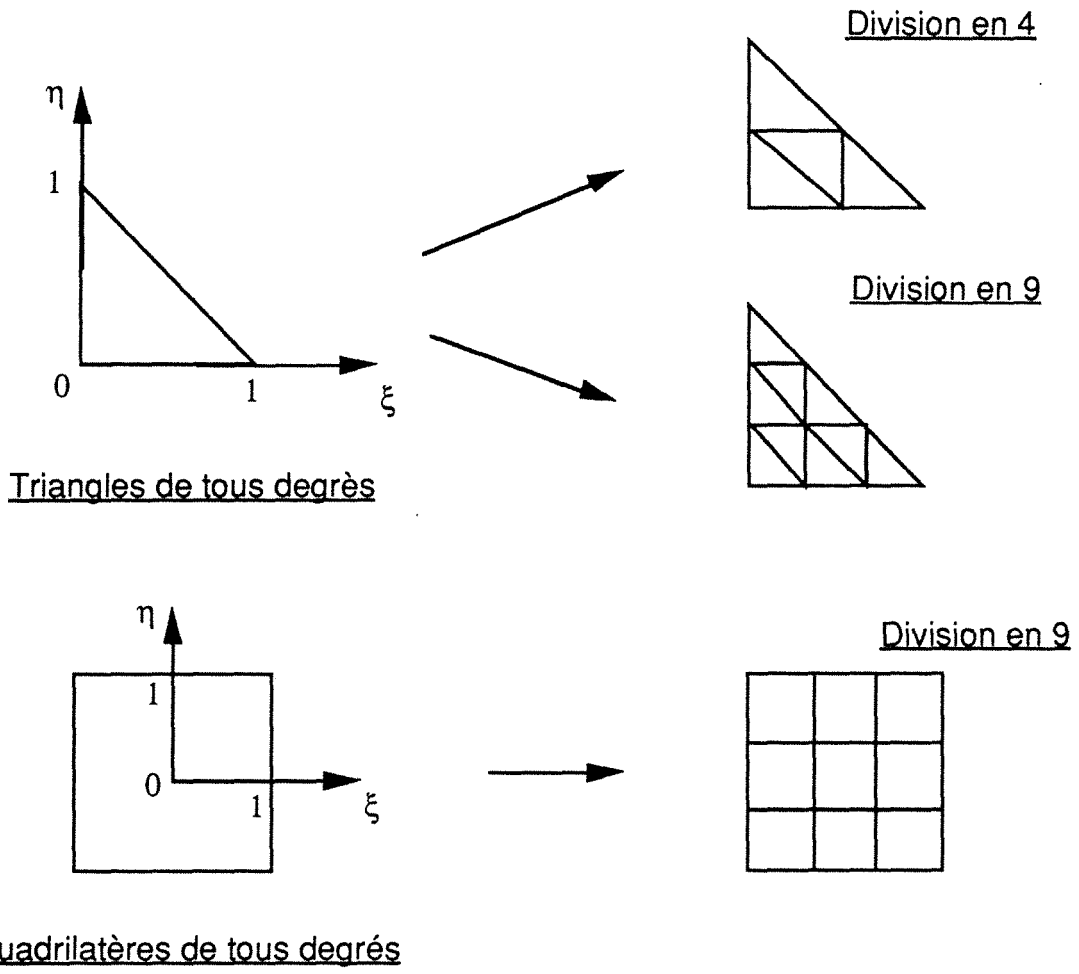
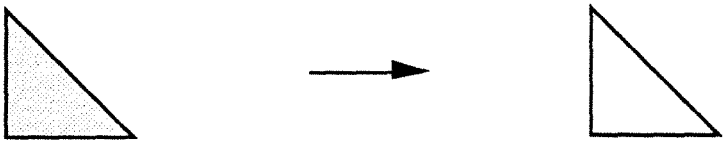
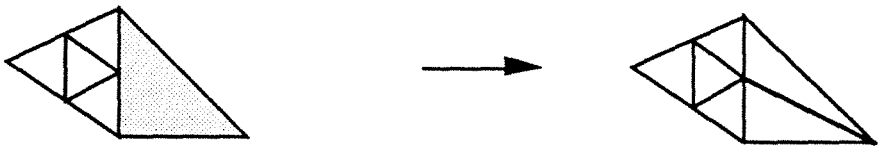


Figure VIII-18 : Division d'éléments

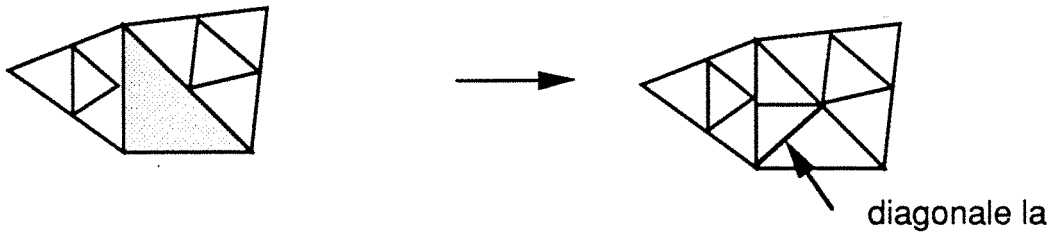
Il reste maintenant à faire le raccord avec le maillage de départ afin d'obtenir un résultat correct. Pour cela, sur chaque type d'élément, nous testons le nombre de voisins subdivisés à l'étape précédente et nous en déduisons, suivant le cas, le travail à faire.

En voici le résultat :

Elémants triangulaires : division en 4

Nombre de voisins divisés et maillage de départ	Solution
zéro voisin divisé	
un voisin divisé	

deux voisins divisés



trois voisins divisés

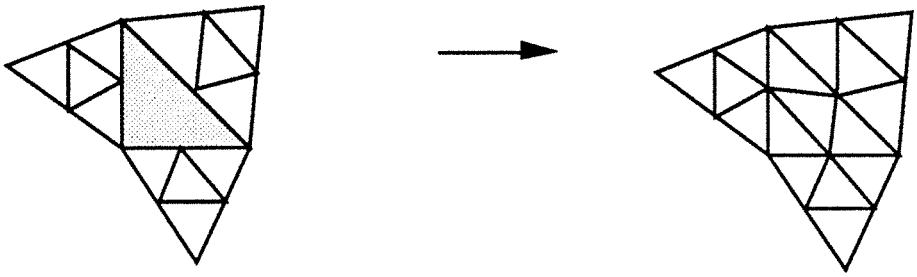


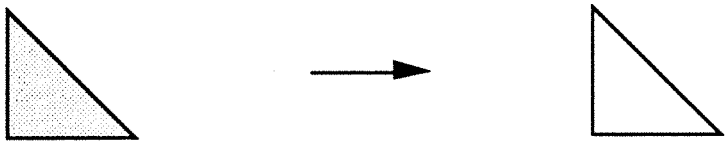
Figure VIII-19 : Eléments triangulaires / division en 4

Eléments triangulaires / division en 9

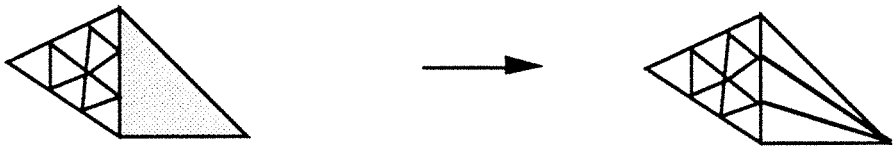
Nombre de voisins divisés
et maillage de départ

Solution

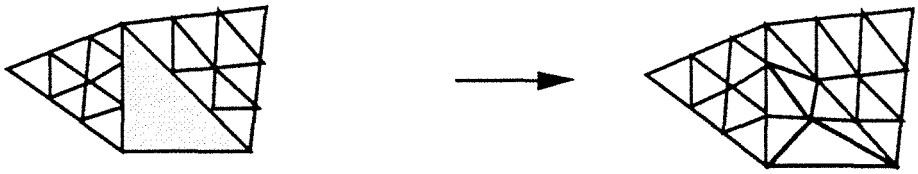
zéro voisin divisé



un voisin divisé



deux voisins divisés



trois voisins divisés

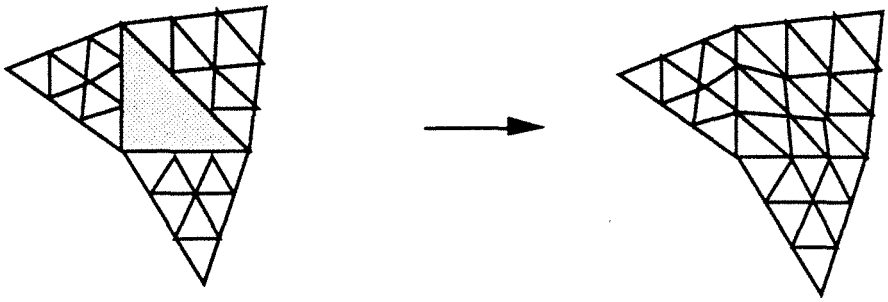


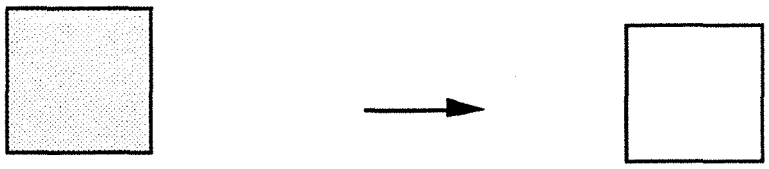
Figure VIII-20 : Eléments triangulaires / division en 9

Eléments quadrangulaires / division en 9

Nombre de voisins divisés et maillage de départ

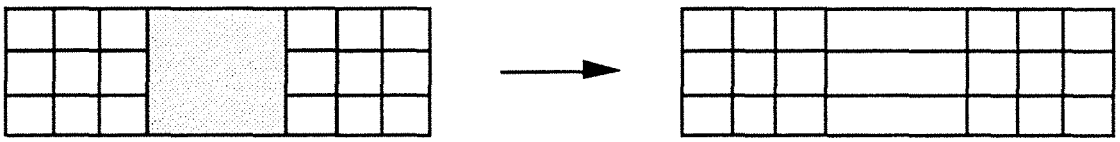
Solution

zéro voisin divisé

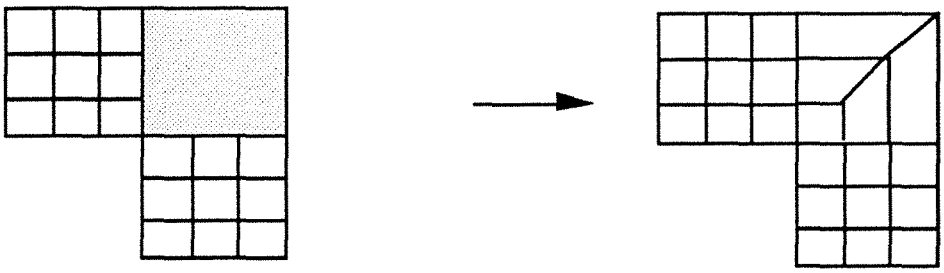


un voisin divisé

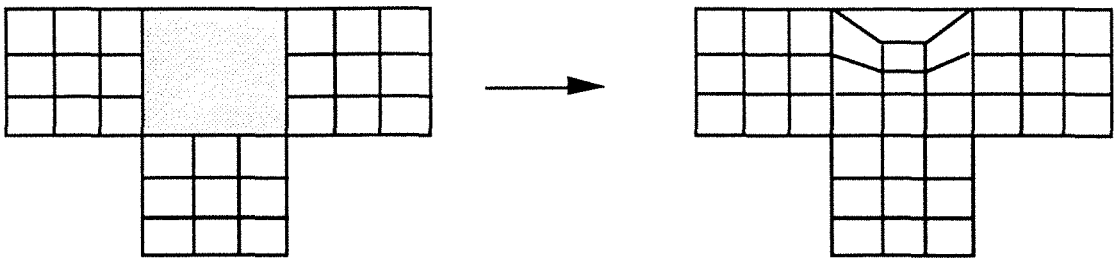
deux voisins divisés (opposés)



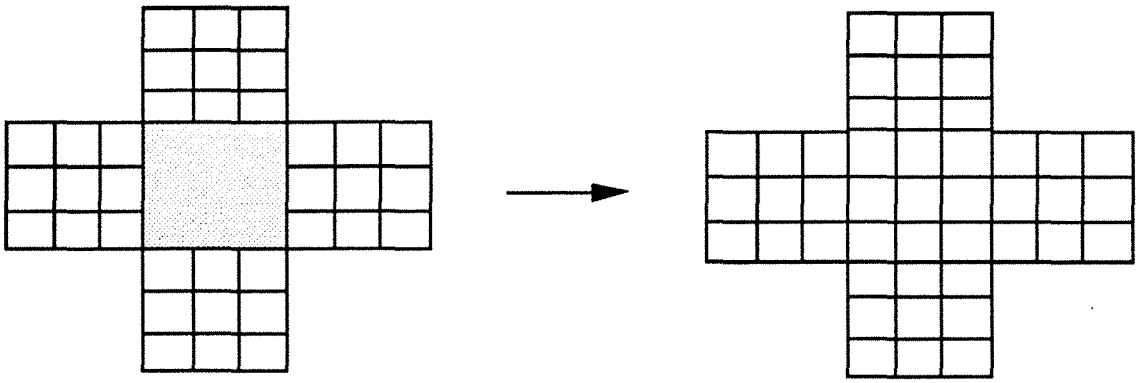
deux voisins divisés (adjacents)



trois voisins divisés



quatre voisins divisés



Toutes ces divisions sont effectuées dans l'élément de référence. Ainsi, les coordonnées de tous les nouveaux points devant être crées sont connues et stockées une fois pour toutes. Le passage dans l'élément réel est alors aisé à effectuer.

Comme précisé au chapitre VI, les diverses grilles sont stockées sur fichier disque et les points en mémoire centrale.

Cette structure permet un passage d'une grille à une autre sans problème particulier même dans les cas les plus complexes (ce qui est souvent le cas lorsqu'on voit les principes de passage d'une grille à l'autre). Cependant, comme c'était prévisible, le temps de calcul pour passer d'une grille à l'autre (restriction ou prolongation) est un peu plus élevé que celui exigé pour un cas de division de grilles uniforme.

VIII-6 : Méthodes multigrilles

Mettons en avant les points particuliers de la méthode multigrilles dans le cas de problèmes plans. Tous les aspects qui ne seront pas mentionnés seront alors conforme à ce qui à été vu au chapitre VI.

Il n'y a en fait pas grand chose à rajouter par rapport à une structure multigrilles "standart". Le seul point à préciser concerne la réactualisation des forces appliquées, des déplacements imposés et de la description du contour du maillage en lui-même. Passons ces points en revue.

VIII-6-1 : Projection sur les frontières courbes

Lors de la division du maillage, les éléments linéaires divisés ayant un côté frontière situé sur une frontière courbe sont traités de façon à satisfaire la courbure de cette frontière. Les élément quadratiques ne sont pas traités, ils ont assez de nœuds par côté pour pouvoir traduire correctement le types de frontières proposées (droites ou courbes).

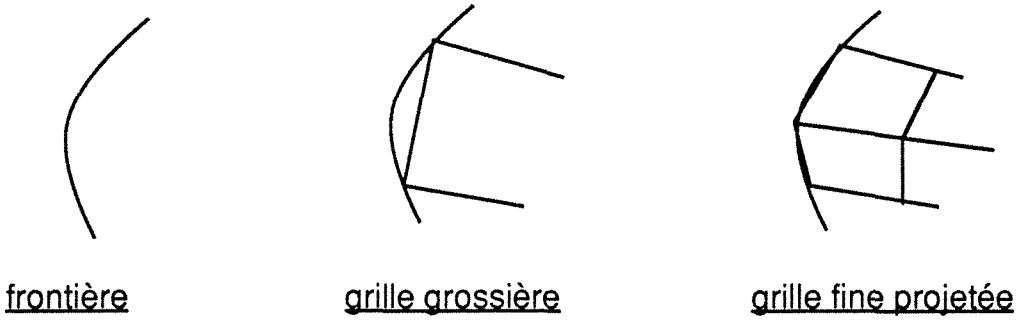


Figure VIII-22 : Projection sur frontière courbe

VIII-6-2 : Réactualisation du contour

Nous avons vu au chapitre I que pour chaque maillage, nous sauvegardions le contour de l'objet formé de la somme des côtés frontières des éléments. La division du maillage et les éventuelles projections font qu'il est nécessaire de redéfinir le contour de la nouvelle grille , ce qui se fait sans problème. Les différents contours de chaque grille sont stockés les uns à la suite des autres sur fichier disque à accés direct. La description de chaque côté frontière se fait de façon conforme à la structure de données vue au chapitre I.

VIII-6-3 : Réactualisation des déplacements imposés

Si tous les points de définition d'un côté frontière de la grille grossière ont leurs déplacements imposés, alors, tous les nouveaux points de la grille fine crées sur ce côté auront leurs déplacements imposés. La valeur de ce déplacement sera interpolé (grace aux fonctions de forme) à partir des valeurs aux nœuds.

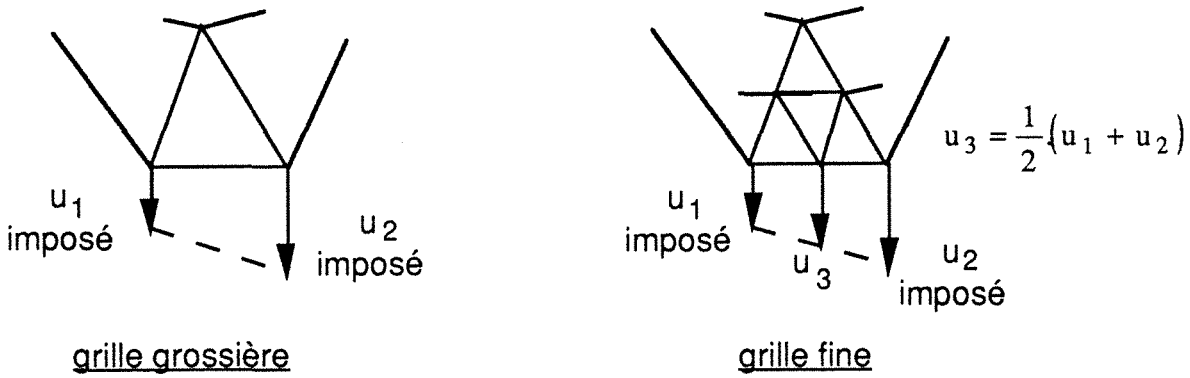


Figure VIII-23 : Réactualisation des déplacement imposés

VIII-6-4 : Réactualisation des forces appliquées

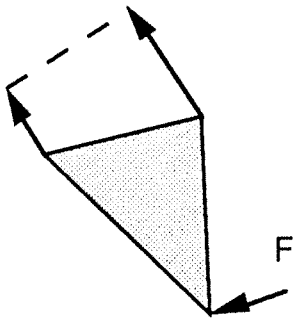
La définition d'une nouvelle grille nécessite la redéfinition du chargement appliqués. Celui ci se divise en trois catégories chacune stockée dans un fichier séparé :

_ les forces ponctuelles ne sont pas modifiées par le changement de grille. La numérotation des points de la grille grossière n'étant pas modifiée, le chargement de la grille fine est donc défini de la même façon que celui de la grille grossière.

_ Les forces surfaciques s'exerçant sur des éléments qui sont divisés ou non sont redéfinies en répartissant celles-ci entre les divers éléments qui sont crées.

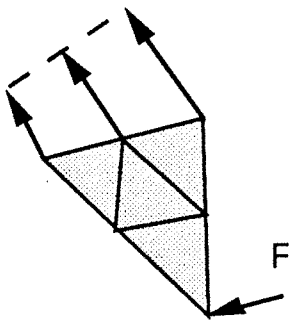
_Les forces linéique s'exerçant sur des côtés d'éléments qui sont divisés ou non sont redéfinies en répartissant celles-ci entre les côtés des divers éléments qui sont crées. La valeur de ces forces sur les nouveaux points crées sont interpolés en fonction des nouvelles valeurs le long des côtés.

Les fichiers contenant les deux derniers types de forces contiennent successivement le chargement de chaque grille, ce qui permet une gestion globale aisée.



grille grossière

_ une force linéique
_ une force ponctuelle F



grille fine

_ deux forces linéiques
_ une force ponctuelle F

Figure VIII-24 : Redéfinition de chargement

VIII-7 : Exemples

VIII-7-1 : Plaque carrée percée

Nous considérons une plaque carrée percée d'un circulaire avec hypothèse des contraintes planes (épaisseur 1mm). Nous observons une concentration de contraintes sur la périphérie du trou à laquelle le maillage d'adapte bien.

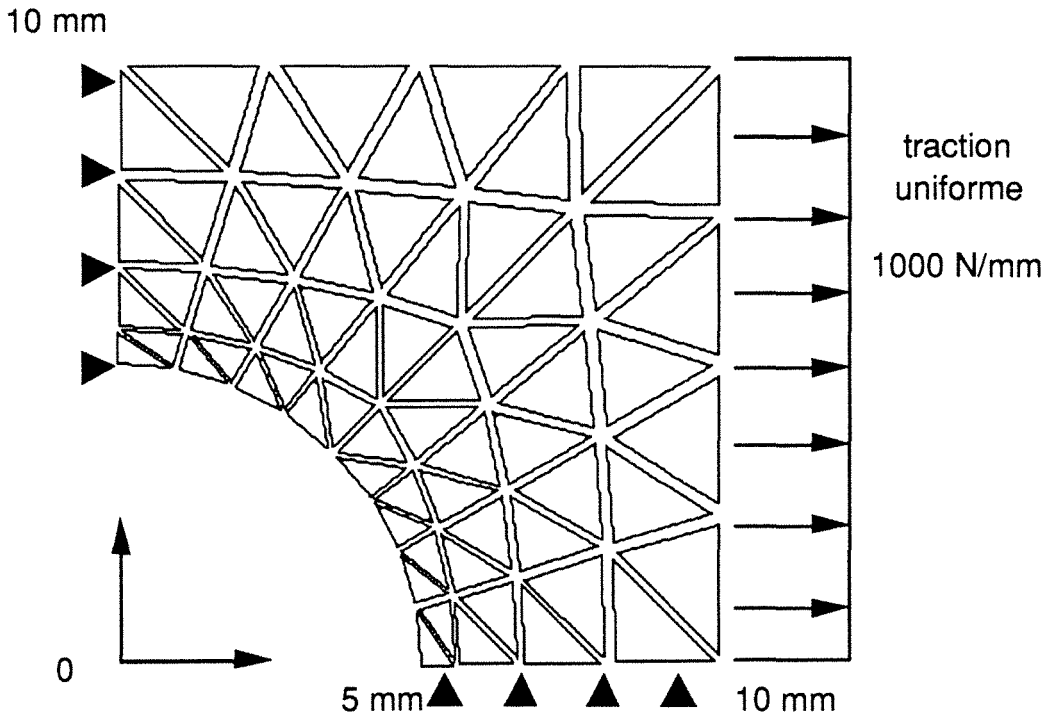


Figure VIII-25-a : Le maillage de départ (grille la plus grossière) 45 nœuds et 64 triangles 3-nœuds

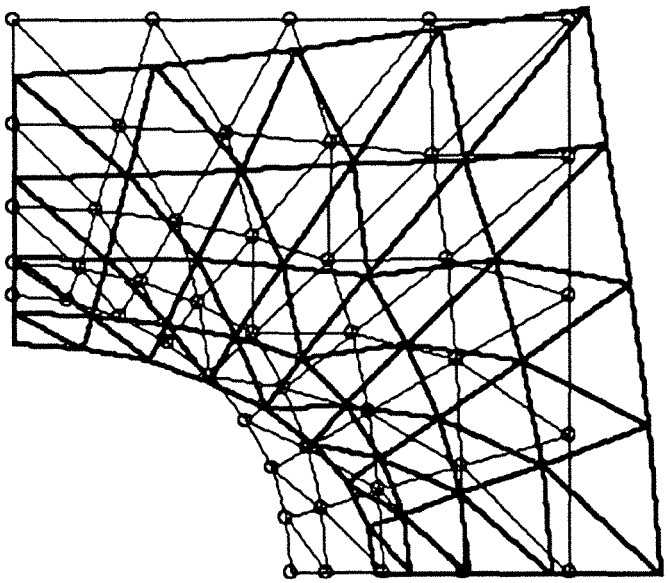


Figure VIII-25-b : La déformée sur la grille grossière (grossie 100 fois) surimprimée avec le maillage non-déformé.

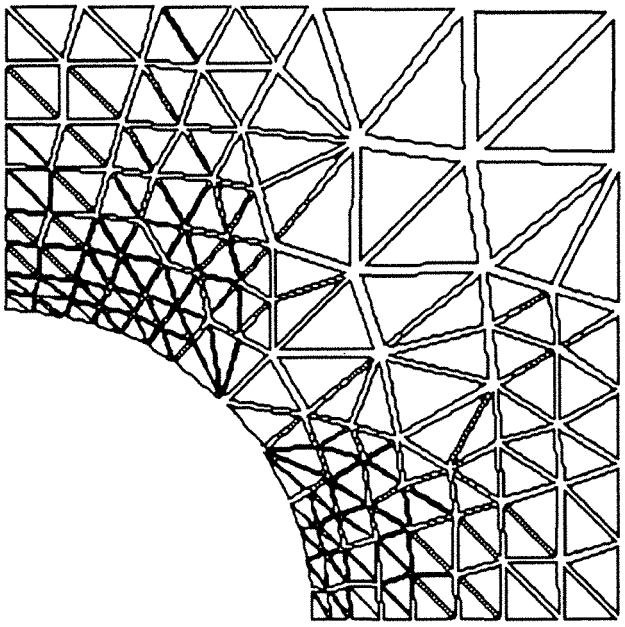


Figure VIII-25-c : La deuxième grille après un raffinement 118 nœuds et 193 triangle 3-nœuds

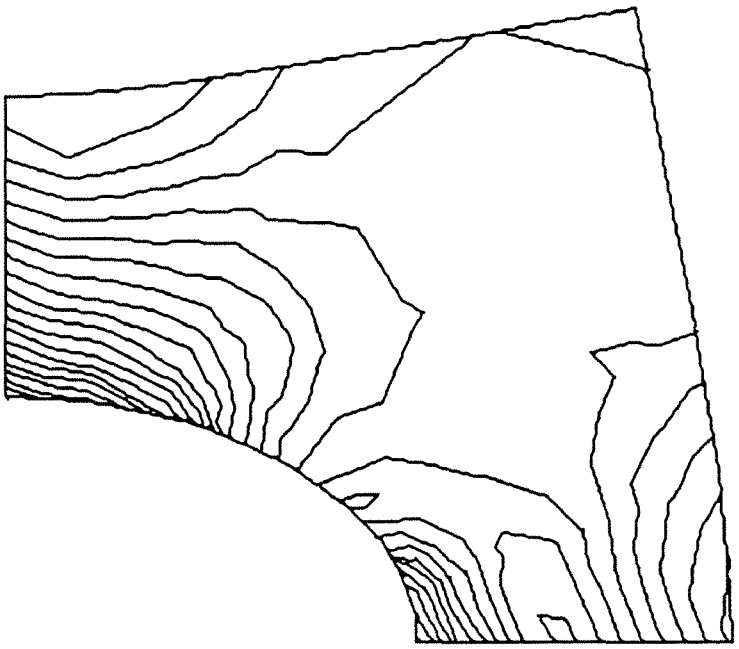


Figure VIII-25-d : Le tracé des isocontraintes de Tresca sur la grille N° 2 après quatre itérations d'une méthode 2-grilles.

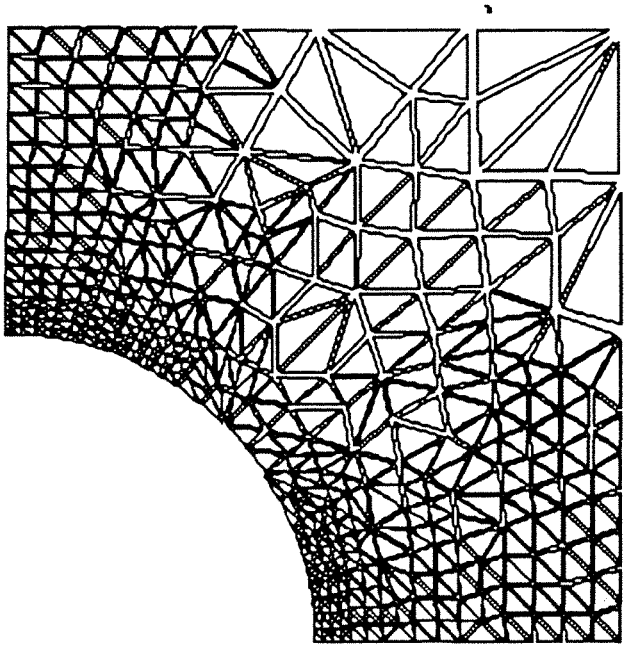
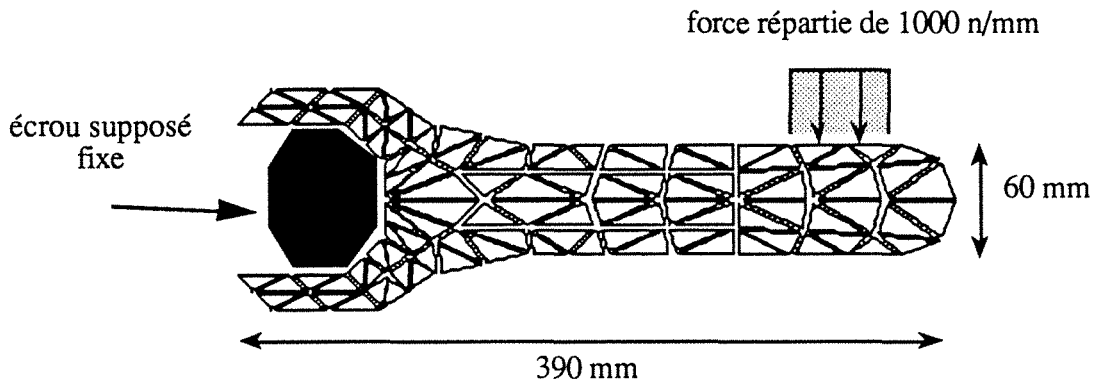


Figure VIII-25-e : La troisième grille est construite à partir de la deuxième (357 nœuds et 638 triangles 3-nœuds)

VIII-7-2 : Etude d'une clé Anglaise

Nous nous intéressons au problème d'une clé Anglaise en flexion correspondant au problème décrit par la figure VIII-26-a :



Hypothèse des contraintes planes (épaisseur de 1 mm)

L'ensemble du matériau est en acier (E=210000 N/mm2)

Figure VIII-26-a : Problème posé et grille grossière de départ (76 nœuds et 104 éléments)

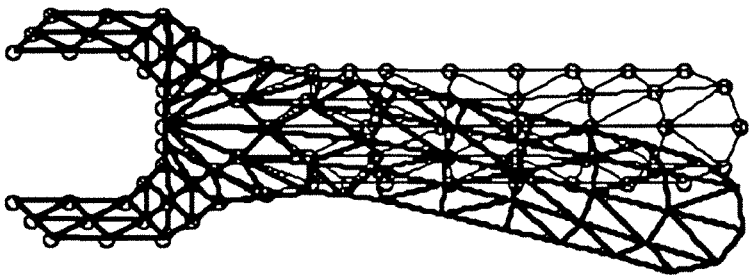


Figure VIII-26-b : La déformée sur la grille N° 1(grossie 100 fois) sur-imprimée avec le maillage

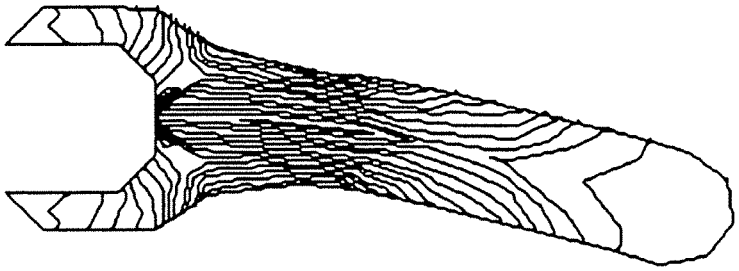


Figure VIII-26-c : Les isocontraintes (critère de Von Mises) sur la grille N° 1

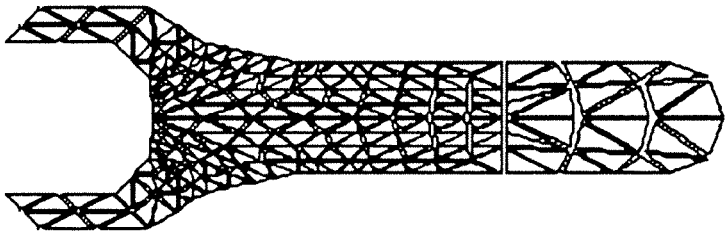


Figure VIII-26-d : Le maillage la grille N° 2 raffiné à partir des résultats précédents (164 nœuds et 264 éléments)

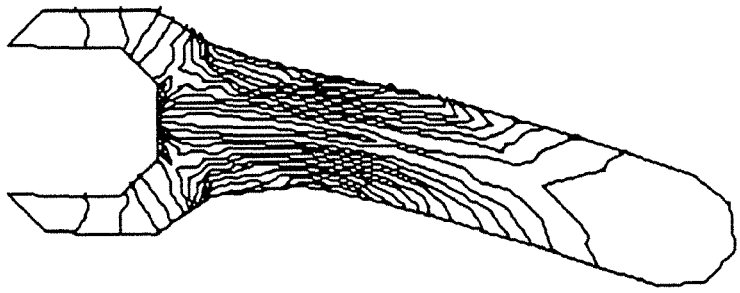


Figure VIII-26-e : Les isocontraintes (critère de Von Mises) sur la grille N° 2 après 6 itérations de la méthode multigrilles entre les grilles 1 et 2

A titre de comparaison, nous avons étudié un maillage uniformément raffiné résolu par une méthode classique pour voir quels étaient les résultats que l'on obtenait :

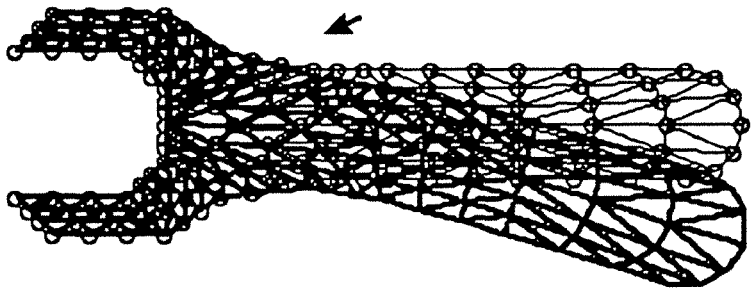


Figure VIII-26-e : Le maillage et la déformée grossie 100 fois

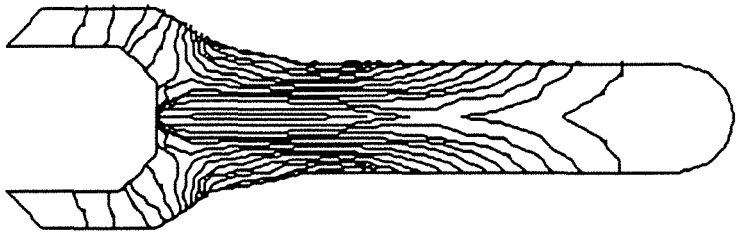


Figure VIII-26-g : Les isocontraintes (critère de Von Mises) sur un maillage uniforme et avec résolution par une méthode classique.

Nous pouvons aussi comparer numériquement les résultats des deux méthodes :

	déplacements maximaux	contraintes maximales selon Von Mises
Grille N° 2 après 6 itérations de la méthode 2-grilles	x max = 0,6804 mm y max = 0,9420 mm	226 MPa
Grille uniforme résolue par une méthode classique	x max = 0,6595 mm y max = 0,9141 mm	215 MPa
écart moyen en %	environ 3 %	environ 7 %

L'erreur relative ne dépasse pas 3 % pour les déplacement et 7 % pour les contraintes. Compte tenu du fait qu'il s'agit de deux maillages différents et que dans le cas d'un problème de flexion traité en contraintes planes, le maillage en triangles influe sur la solution, nous pouvons admettre que nous convergeons vers la même solution.

L'écart le plus important concerne les contraintes, mais nous rappelons que celle-ci sont lissées par moindre carré et que le résultat dépend donc du maillage plus nettement que dans le cas des déplacements.

VIII-7-3 : Etude d'une plaque carrée encastrée

Nous nous intéressons au problème d'une plaque carrée (100 mm de côté) mince encastréé en traction avec hypothèse des contraintes planes (épaisseur de 1 mm) coorespondant à la figure VIII-27-a

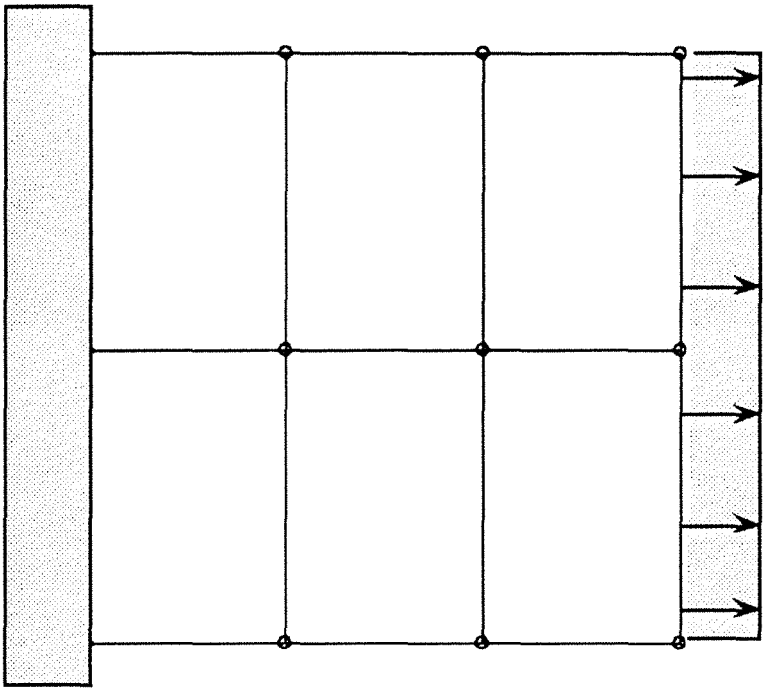


Figure VIII-27-a : Problème posé et Grille N° 1 (6 éléments 4-nœuds et 9 nœuds)

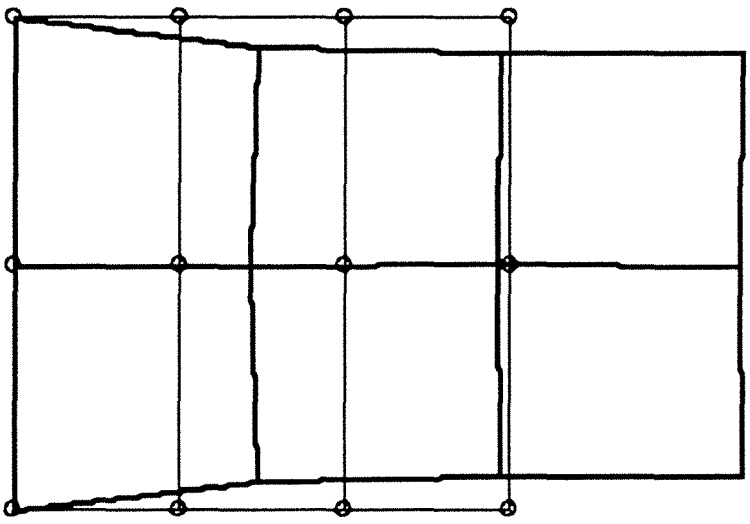


Figure VIII-27-b : Le maillage et sa déformée de la grille N°1 grossie 100 fois

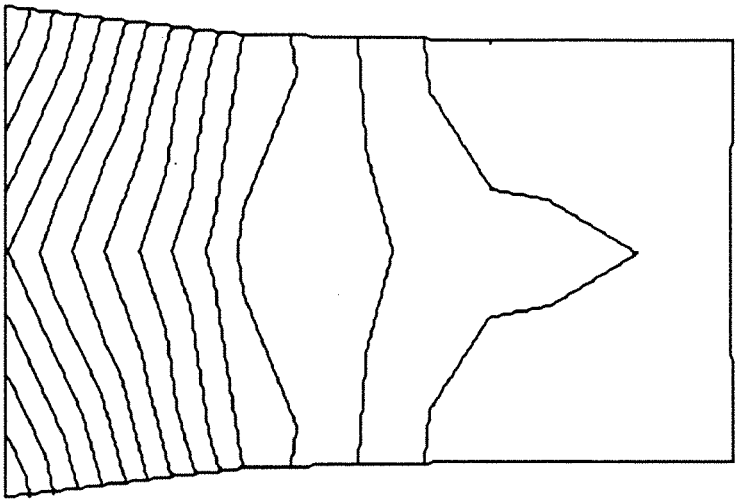


Figure VIII-27-c : Les isocontraintes sur la grille N°1 (critère de Von Mises)

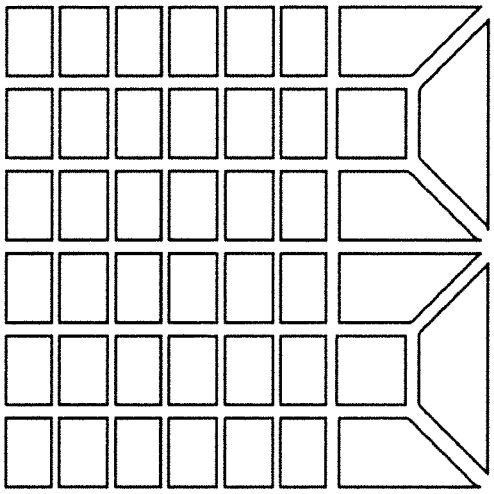


Figure VIII-27-d : La grille N°2 Calculée à partir des résultats précédents

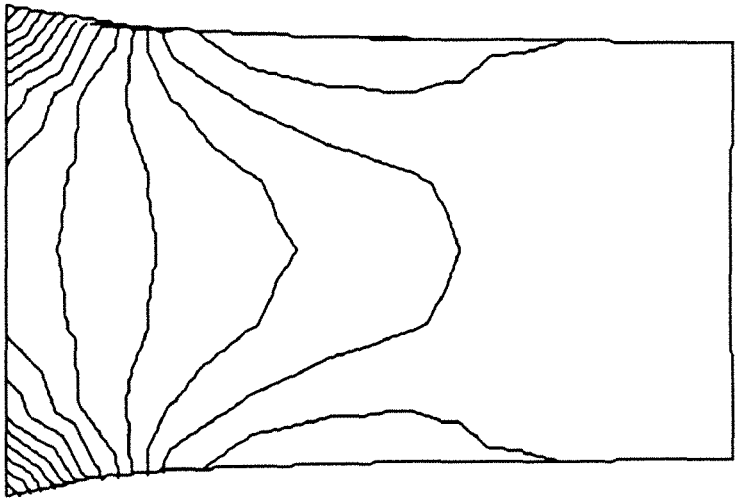


Figure VIII-27-e : Les isocontraintes sur la grille N°2 après 4 itérations de méthode 2-grilles

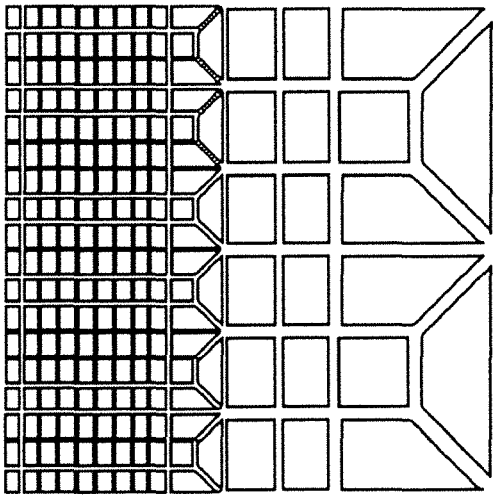


Figure VIII-27-f : La grille N°3 Calculée à partir des résultats précédents

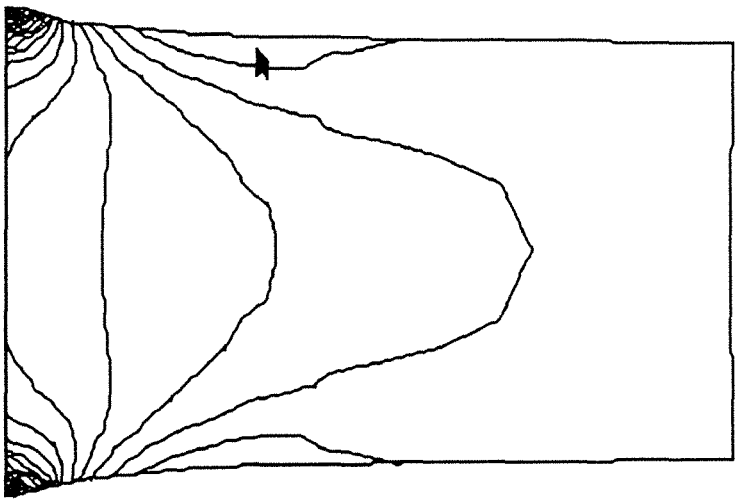


Figure VIII-27-g : Les isocontraintes sur la grille N°3 après 4 itérations de méthode 3-grilles

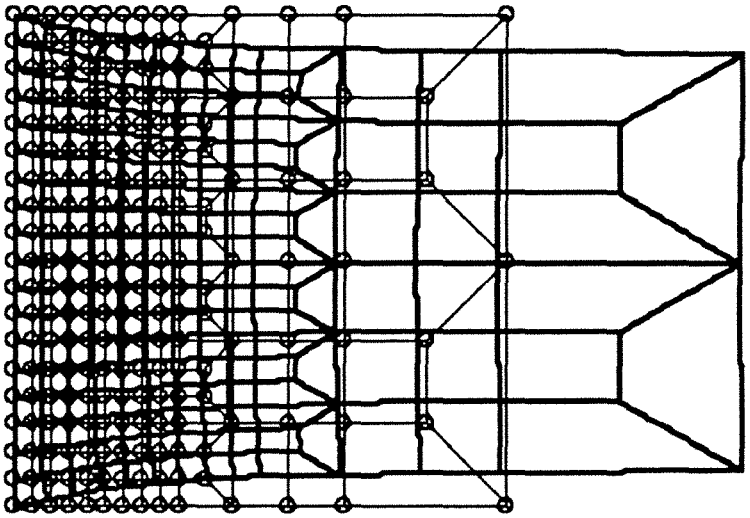


Figure VIII-27-h : Le maillage et sa déformée de la grille N°3 grossie 100 fois

valeurs test	grille N° 1	grille N° 2	Grille N°3	Références **
déplacements maximums en mm	u = 0,4696 v = 0,07391	u = 0,4715 v = 0,07412	u = 0,4721 v = 0,07400	u = 0,4723 v = 0,074
contraintes maximales en N/mm2 (von Mises)	σ max= 2017 σ min = 1716	σ max= 2278 σ min = 1667	σ max= 2789 σ min = 1619	σ max= 2837 σ min = 1656

Tableau comparatif entre la méthode Full multigrilles 3-grilles et un maillage raffiné manuellement (par un utilitaire inclus dans le mailleur de Delaunay) et résolu par une méthode classique.

** est un maillage de Delaunay en triangle 3-nœuds raffiné manuellement et localement par un utilitaire inclus dans le mailleur de Delaunay et comptant 182 éléments et 110 nœuds (figure VIII-28).et résolu par une méthode classique (SSOR CG).

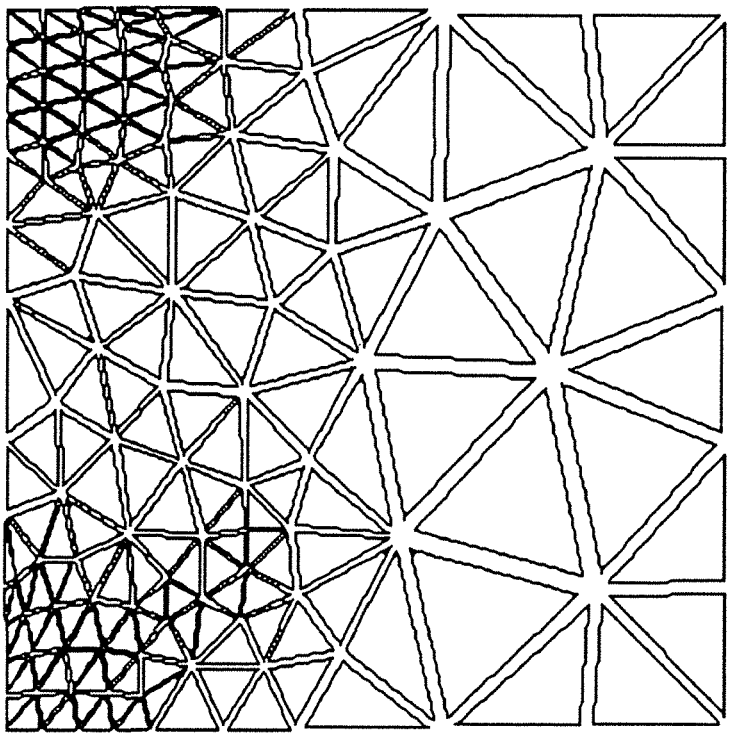


Figure VIII-28 : Le maillage de référence (182 éléments et 110 nœuds)

Nous constatons que le processus multigrille converge bien vers nos valeurs de références. Cependant, la convergence est plus nette sur les contraintes que sur les déplacements. ceci tend à valider notre processus de maillage adaptatif qui fournit de meilleurs valeurs de contraintes dans le cas de concentrations de contraintes.

VIII-7-4 : Etude d'un quart d'anneau en compression

Nous nous intéressons au problème d'un anneau d'acier en compression avec hypothèse des contraintes planes, dont nous n'étudions qu'un quart par raison de symétrie (Figure VIII-29-a).

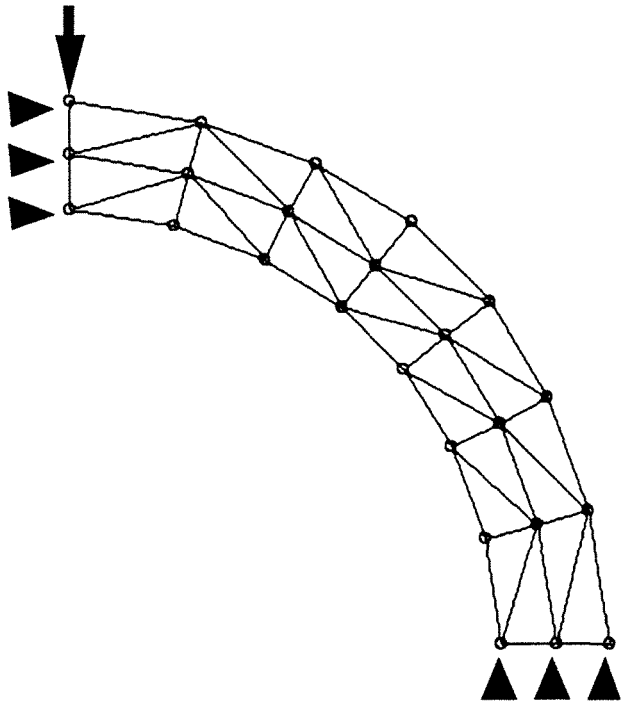


Figure VIII-29-a : Problème posé et Grille N° 1 (28 éléments 3-nœuds et 32 nœuds)

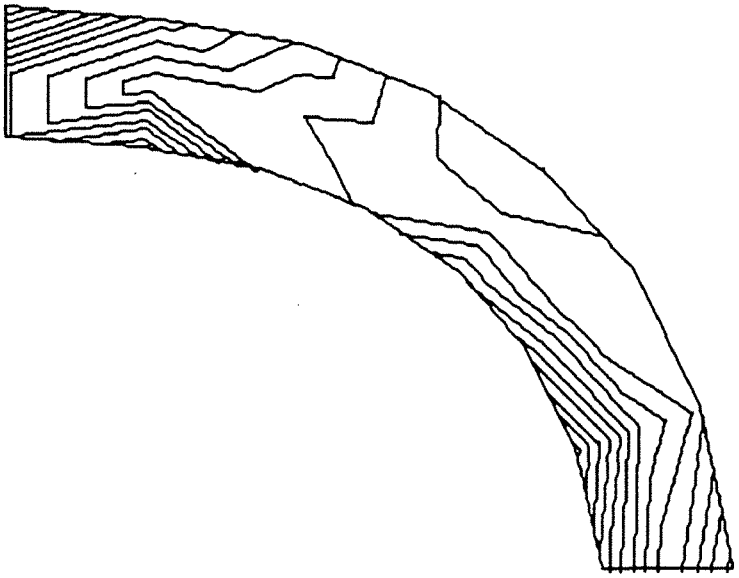


Figure VIII-29-b : Les isocontraintes sur la grille N°1 (on notera les mauvaise valeurs, surtout au niveau des axes de symétrie)

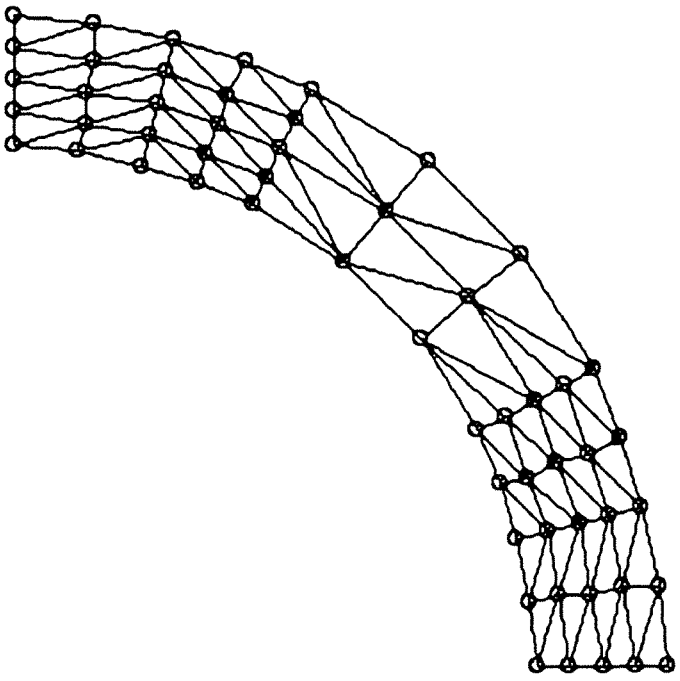


Figure VIII-29-c : La grille N°2 Calculée à partir des résultats précédents

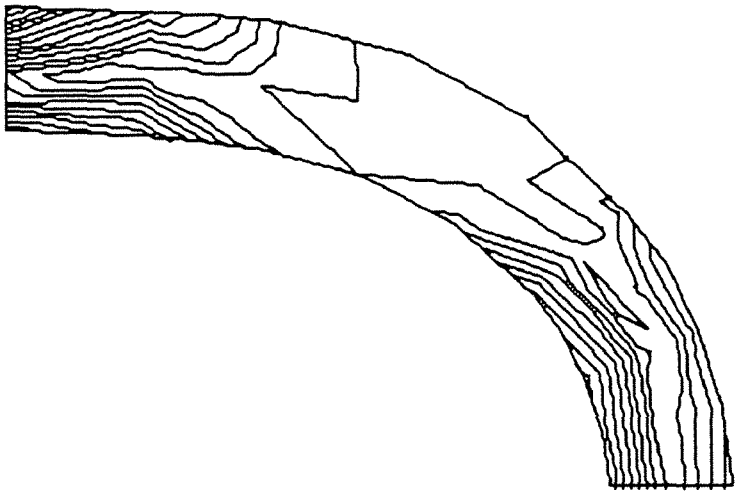


Figure VIII-29-d : Les isocontraintes sur la grille N°2 après 6 itérations d'une méthode 2-grilles: la répartition des contraintes est bien meilleure, surtout au niveau des axes de symétrie (cf fig 29-b).

Dans ce cas (et dans les cas généraux de flexion), l'élément triangulaire 3-nœuds est assez lent à converger. Cet aspect se retrouve dans le processus multigrilles pour lequel on retrouve de fortes variation de déplacements (et a fortiori de contraintes) d'une grille à une autre. Dans ce cas, il est nécessaire d'itérer plus longuement pour avoir une valeur correcte.

VIII-8 : Conclusions

Nous venons de voir que la méthode multigrille a prouvé son efficacité dans le cas bidimensionnel. Il semble bien que celle-ci ,couplée avec un calcul d'erreur et un enrichissement sélectif du maillage (par h- ou p-raffinement) constitue une amélioration judicieuse des méthodes d'éléments finis pour les systèmes de taille importante. Cette question fait d'ailleurs l'objet de recherches dans certains laboratoires Français et Etrangers [IN 3], [IN 5], [IN 18] et [EL 18].

Du point de vue de l'efficacité, il semble dans notre cas que le maillage en triangles conduise à des résultats les plus performants. Ceci n'est pas une surprise : le processus de division des quadrilatères crée des éléments parfois assez distordus, surtout après plusieurs divisions de grilles. Par ailleurs, le passage d'une grille à une autre est un peu plus "abrupt" dans le sens où le nombre d'éléments crée est plus important (division en 9 seulement) , et l'on sait que ceci n'avantage pas les méthodes multigrilles (voir chapitre V). La nature du type d'enrichissement de grille choisi (h-raffinement) est donc très important; il est vraissemblable qu'un autre procédé donnerait de meilleures résultats.

Les éléments triangulaires n'ont pas ce genre d'états d'âme. La division d'éléments triangulaires est beaucoup plus souple et efficace, et elle crée des éléments moins distordus. On notera cependant que le triangle 3-nœuds est un peu faible pour ce genre d'application. Nous suggerons donc l'utilisation du triangle 6-nœuds. Il constitue un élément performant et il assure une interpolation complète des termes du second degré, propriété qui le rend très stable vis à vis de la distorsion , donc bien adapté au h-raffinement. Utilisé sous sa forme isoparamétrique, cet élément peut être utilisé par des mailleurs automatiques (comme Delaunay) et peut décrire sans problème les domaines les plus variés.

Comme c'était prévisible, les opérations de transfert d'une grille à une autre prenent beaucoup de temps. De ce point de vue là, le principe d'enrichissement sélectif du maillage n'arrange pas les choses. On notera en particulier que les phases de redéfinition du maillage , des matériaux, des forces et du contour sont assez coûteuses, et ce principalement parce que l'enrichissement sélectif du maillage conduit à de nombreux cas particuliers. On peut donc prévoir qu'un traitement identique en 3D (si c'est possible) augmenterait assez fortement le temps de calcul et serait tout bonnement abominable à programmer tant les cas particuliers pulluleraient.

Il n'y a cependant aucun doute possible, l'enrichissement sélectif du maillage demeure tout à fait indispensable. Ce que l' on perd en temps de transfert est largement compensé par la diminution de la taille des grilles (par rapport à un enrichissement homogène de la grille), des vecteurs et surtout de la matrice de rigidité. Cette optique est bien adapté au travail sur micro-ordinateur pour lequel la mémoire fait parfois défaut.

L'utilisation de la méthode Full-multigrilles avec enrichissement selectif du maillage semble donc être ce qui s'adapte le mieux aux exigences du travail sur micro-ordinateur. Il est vraissemblable qu'on puisse améliorer les performances de la méthode par un ajustement plus judicieux des composantes, mais cet ajustement reste délicat car le comportement théorique de la méthode n'est pas maitrisé dans le cas général.

Conclusions

Notre conclusion, respectant le plan de ce mémoire, va se décomposer en deux parties, l'une traitant des problèmes de maillage, l'autre traitant des problèmes de résolution par les méthodes multigrilles.

Les logiciels de maillage que nous avons étudiés et développés nous ont donné entière satisfaction. Ils ont été testés avec succès sur des domaines complexes décomposés en centaines d'entités.

Leur facilité d'emploi, leur aspect général et le nombre assez varié d'éléments utilisables permet leur utilisation dans des applications diverses. Ainsi, les maillages crées servent-ils de supports à des études numériques de problèmes de mécanique des fluides et d'homogénéisation.

La structure de données que nous avons choisie permet de décrire complètement le domaine à mailler ainsi que le maillage conçu, et ce indépendamment du problème, ce qui permet un plus grand découplage entre les mailleurs et les processus de modélisation et de résolution. Ainsi, par exemple, la description précise du contour d'un domaine peut-il servir à reconnaître et à affecter des conditions limites particulières dans le cas de problèmes d'homogénéisation.

En ce qui concerne les algorithmes de maillage, il semble que le plus efficace _ du point de vue de l'automaticité _ soit celui de Delaunay., associé à un processus de forçage d'arêtes. Cependant, il n'offre qu'un maillage en éléments triangulaires, ce qui peut être un inconvénient dans le cas où les éléments quadrilatéraux sont imposés (certains problèmes de plasticité par exemple). l'élaboration d'un algorithme de maillage en quadrilatères totalement automatique est encore du domaine de la recherche. Le problème est d'autant plus ardu à résoudre qu'il existe des cas où il n'y à pas de solution. Dans cette optique, l'utilisation d'un mailleur semi-automatique par bloc complété par une gamme étendue d'utilitaires de post-traitement permet à moindre frais de résoudre le problème.

De ce point de vue, les méthodes que nous avons développées répondent bien aux exigences de maillages de domaines 2D. Nous continuons cependant à étudier et à développer des méthodes nouvelles, notamment pour les quadrilatères, permettant d'améliorer les performances de maillage.

Le problème est plus complexe en 3D. D'abord parce que la représentation en perspective rend difficile la définition des pièces et leur visualisation, ce qui nécessite l'intervention d'utilitaires coûteux en temps de calcul (coupes, projections, élimination des parties cachées). Enfin, parce que les algorithmes développés, s'ils sont efficaces, sont moins automatiques.que ceux développés en 2D. Ces contraintes imposent ici une automatisation poussée de la description de la pièce (contour + maillage) pour garder une certaine efficacité.

Nous adaptons actuellement une méthode de Delaunay associée à un processus de forçage de faces, au cas tridimensionnel, la délicate description du contour étant effectuée par interpolation (fonctions splines ou de Bézier). Pour être efficace, un tel processus nécessitera l'utilisation d'une structure de données "orientée objet" (au sens informatique du terme). Il faudra donc utiliser un langage sachant concilier puissance numérique et souplesse informatique : dans l'état actuel des choses, vraisemblablement le langage C++.

En ce qui concerne la résolution, la méthode multigrilles appliquée au cas de l'élasticité linéaire plane ainsi qu'aux ossatures de poutres planes ont donnés des résultats tout à fait conforme à ceux que nous laissaient présager la théorie. Cependant, comme c'était prévisible, l'efficacité de la méthode multigrilles dépend du bon ajustement des diverses fonctions qui la composent (type de cycle, lissage, restriction, résolution sur grille grossière, etc...).

Ce bon ajustement dépend de deux critères donnés :

_ L'efficacité de chaque composant choisi.

_ Le coût de calcul et de place mémoire.

En ce qui concerne le premier point, nous avons pu noter les faits suivants :

_ La version le plus efficace de la méthode multigrilles est la version "Full-Multigrilles". L'aspect dynamique de cette méthode, surtout couplé avec un processus de remaillage adaptatif, lui confère une efficacité remarquable. L'utilisation du W-cycle offre un bon rapport efficacité-temps de calcul.

_ Le lissage par un processus de Gauss Seidel ou de Jacobi donne de bons résultats à moindres frais (place mémoire et temps de calcul). En général, entre 3 et 5 itérations par niveaux suffisent.

_ L'utilisation des fonctions de forme pour définir prolongation et restrictions donnent entièrement satisfaction, nous les avons donc utilisés systématiquement chaque fois que cela était nécessaire, particulièrement dans le cas du processus Full multigrilles.

_ Enfin, la résolution sur grille grossière par une méthode SSORCG est tout à fait efficace si le problème est de taille quelconque (jusqu'à quelques centaines d'inconnues). Pour les systèmes de tailles plus réduites, nous développons une méthode Frontale qui permettra de réduire les temps de calculs ainsi que la place mémoire nécessaire.

En ce qui concerne le dernier point, nous pouvons penser que :

_ Le stockage des informations, tel qu'il est conçu, apporte bien des facilités pour utiliser les méthodes multigrilles, surtout en 2D. Il pourra cependant être modifié dans le cas d'applications précises apportant des contraintes particulières (problèmes d'homogénéisation par exemple).

_ L'utilisation de la méthode Morse pour compacter efficacement les matrices de rigidité permet d'éviter d'avoir à renuméroter les points sur chaque grille, il apporte donc un gain de place et de temps de calcul intéressant.

Sur un plan général, nous estimons que pour obtenir l'efficacité optimale, la méthode doit répondre aux impératifs suivants :

_ L'élément fini utilisé doit être assez performant pour traiter le type de problème souhaité.

_ Le nombre de grilles ne doit pas être trop élevé (entre 2 et 5) pour ne pas ralentir la méthode par des redéfinitions de grilles et des transferts d'informations coûteux en temps de calcul.

L'utilisation d'un processus de résolution bidimensionnelle avec maillage auto-adaptatif semble être une façon efficace d'utiliser les méthodes multigrilles. A cet égard, l'utilisation d'un seul type d'élément est un peu restrictif, notamment pour les éléments quadrilatéraux. La solution pourra être de panacher les types d'éléments (par exemple intercaler quelques éléments triangulaires dans un maillage en quadrilatères) afin de pouvoir disposer de maillages plus simples et donc de diminuer les coûts de calcul et la place mémoire occupée. On pourra certainement améliorer les performances de la méthode par l'utilisation de bases hiérarchiques permettant d'économiser du temps de calcul lors de la redéfinition des matrices de rigidité.

Nous prolongerons naturellement ce travail par l'étude de problèmes non-linéaires avec utilisation de la méthode FAS, ainsi que l'étude de problèmes tridimensionnels.

L'étude et le développement systématique de la méthode multigrilles sur micro-ordinateur n'est pas simple : sa complexité et sa sensibilité en font un outil parfois difficile à maîtriser mais dont l'étude est instructive. La réalisation de logiciels de maillage nous a conduit à nous intéresser à des domaines variés et étendus : graphisme, algorithmique etc.... Enfin, la mise au point de logiciels interactifs et conviviaux nous a aura imposé un important effort de clarification et de rigueur. Cela nous aura donc permis d'aborder des domaines très variés : Analyse numérique, Programmation, Graphisme, Architecture de programme.

Annexe I : Fonctions de forme de l'élément isoparamétrique 8-nœuds

les 8 fonctions de forme de cet élément sont exprimées dans le repère de référence de l'élément comme indiqué sur le figure AI-1 ci-dessous :

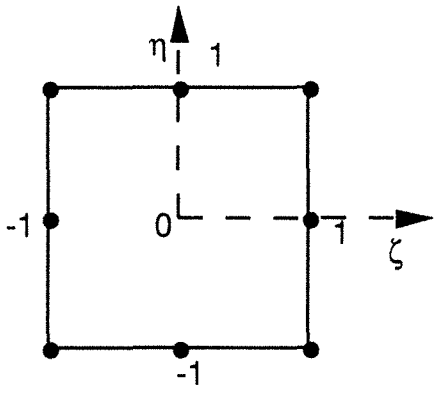


Figure AI-1 : Repère de référence de l'élément 8-nœuds

Les 8 fonctions sont les suivantes

$$N_1(\xi,\eta) = \frac{1}{4}.(1-\xi).(1-\eta).(-1-\eta-\xi)$$

$$N_2(\xi,\eta) = \frac{1}{2}.\left(1-\xi^2\right).(1-\eta)$$

$$N_3(\xi,\eta) = \frac{1}{4}.(1+\xi).(1-\eta).(-1+\eta-\xi)$$

$$N_4(\xi,\eta) = \frac{1}{2}.\left(1-\eta^2\right).(1+\xi)$$

$$N_5(\xi,\eta) = \frac{1}{4}.(1+\xi).(1+\eta).(-1+\eta+\xi)$$

$$N_6(\xi,\eta) = \frac{1}{2} . \left(1-\xi^2\right) . (1+\eta)$$

$$N_7(\xi,\eta) = \frac{1}{4} . (1-\xi) . (1+\eta) . (-1+\eta-\xi)$$

$$N_8(\xi,\eta) = \frac{1}{2} . \left(1-\eta^2\right) . (1-\xi)$$

Annexe II : Maillage de Voronoï et de Delaunay

Le processus de Voronoï est basé sur un ensemble de données absolument identique à celles de Delaunay . Nous nous plaçons dans le cas de la dimension 2 pour fixer les idées et faciliter la représentation des figures, cependant, la méthode développée est valable en dimension p quelconque, ce qui constitue sa grande force.

Considérons $P_1,\ldots,P_n$, n points donnés d'un espace E donné. A tout point P_i , on associe un polyèdre V_i dit "polyèdre de Voronoï" déterminé de la façon suivante :

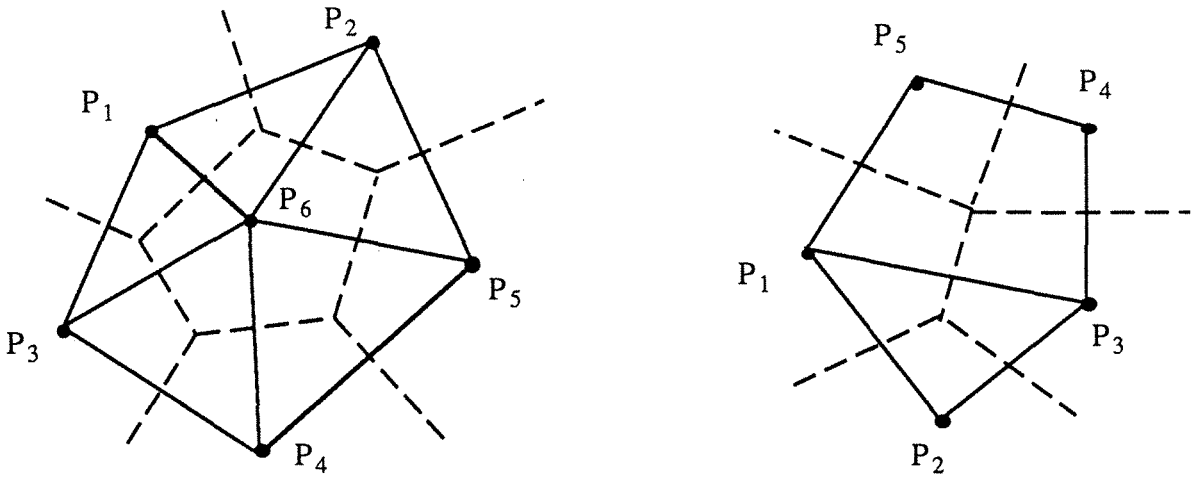
$$V_i = \left\{ x \in E \ / \ \forall j ; \ 1 \le j \le n \ , \ d(x,x_i) \le d(x,x_j) \right\}$$

Il s'agit donc de la zone "la plus proche de P_i que de tout autre point" . On démontre que l'ensemble des polyèdres de Voronoï V_i sont convexes (bornés ou non) et constituent un maillage V (au sens des éléments finis) de l'espace E appelé "maillage de Voronoï" .

Si nous supposons que les n points ne sont pas tous éléments d'un méme hyperplan, alors il est possible de caractériser l'ensemble des polyèdre de Voronoï par leurs sommets $Q_1, \ldots, Q_p$.

On construit alors, pour chaque Q_i , l'enveloppe convexe D_i basée sur ceux des points P_i qui admettent Q_i comme sommet du polyèdre de Voronoï associé . Chaque D_i est un polyèdre convexe et borné . L'ensemble de ces D_i constitue un maillage D dit "de Delaunay" de l'enveloppe convexe des points $P_1,\ldots,P_n$. Ce maillage est le dual (au sens de la théorie des graphes) du maillage de Voronoï V. Selon [IN 1], le maillage de Delaunay sera dit "spécial" s'il contient au moins un élément qui n'est pas un triangle . Dans ce cas, nous appelerons triangulation de Delaunay toute triangulation déduite du maillage de Delaunay par division en triangles des éléments de D qui ne le sont pas . Dans le cas où le maillage n'est pas "spécial", le maillage D constitue déjà une triangulation de Delaunay.

On trouvera en figure AII-1, deux exemples de maillages de Voronoï et Delaunay .



Maillage normal

Maillage "spécial"

— — — — : maillage de Voronoï

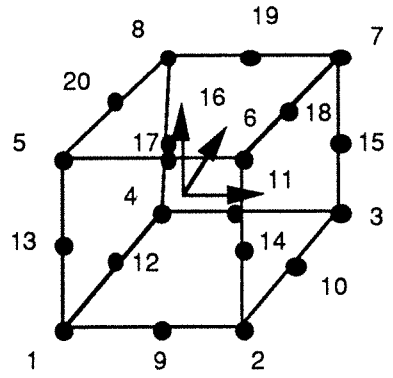
——— : maillage de Delaunay

Figure AII-1 : Maillages de Voronoï et Delaunay

L'une des grandes forces de cette méthode est sa validité à toutes les dimensions. On en trouvera dans [IN 1] une étude générale détaillée. Ceci permet de définir une méthode générale de maillage automatique trés performante, et utilisée intensivement en 2D et en 3D . Dans ce derniers cas, elle constitue l'une des rares méthodes totalement automatique existante. le seul inconvénient que l'on pourra noter réside dans le fait que l'on ne peut, au départ, mailler que dans un seul type d'éléments (triangles en 2D, tétraèdres en 3D) . L'utilisation d'autres types d'éléments nécessite un post-traitement qui n'est pas toujours évident ni même efficace .

Annexe III : Fonctions de forme de l'élément isoparamétrique 20-nœuds

L'élément de référence est représenté dans le repère de référence (ξ,η,ζ).(figure AIII-1). Nous noterons (ξ_i, η_i, ζ_i) (i=1,...,20) les coordonnées de chacun des 20 nœuds définissant l'élément (attention à la numérotation).



élément de référence (ζ, η, ξ)
cubique, centré sur 0 et de côté 2

Figure AIII-1 : Repère de référence de l'élément 20-nœuds

Dans ce repère, les fonction de formes sont les suivantes :

_ Pour les 8 nœuds sommets :

$$N_i(\xi,\eta,\zeta) = \frac{1}{8} . (1+\xi.\xi_i) . (1+\eta.\eta_i) . (1+\zeta.\zeta_i) . (\xi.\xi_i + \eta.\eta_i + \zeta.\zeta_i - 2) \quad i=1,...,8$$

_ Pour les 12 nœuds milieux :

$$N_i(\xi,\eta,\zeta) = \frac{1}{4}.(1-\xi^2).(1+\eta.\eta_i).(1+\zeta.\zeta_i) \quad i=9,11,17,19$$

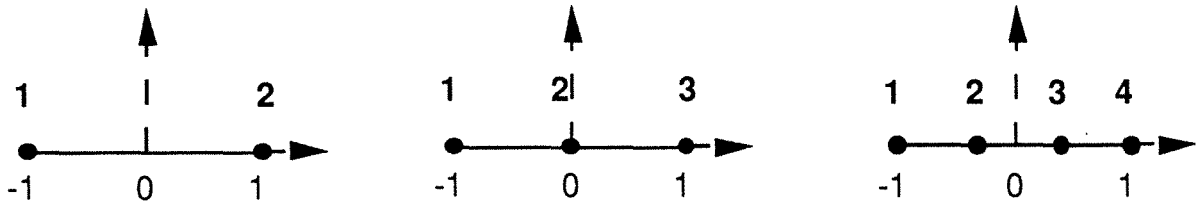
$$N_i(\xi,\eta,\zeta) = \frac{1}{4}.(1-\eta^2).(1+\xi.\xi_i).(1+\zeta.\zeta_i) \quad i=10,12,18,20$$

$$N_i(\xi,\eta,\zeta) = \frac{1}{4}.(1-\zeta^2).(1+\xi.\xi_i).(1+\eta.\eta_i) \quad i=13,14,15,16$$

Annexe IV : Fonction de formes des éléments
de Lagrange et d'Hermite utilisés
pour la résolution de problèmes de poutres

Eléments de Lagrange (continuité C^0) :

Nous utilisons les éléments de Lagrange à 2, 3, et 4 nœuds représentés sur la figure AIV-1 ci dessous. La numérotation des nœuds figure en caractères gras.



Lagrange 2-nœuds Lagrange 3-nœuds Lagrange 4-nœuds

Figure AIV-1 : repères de référence des éléments de Lagrange

Nous obtenons les fonctions de forme suivantes, exprimées dans le repère de référence :

Lagrange 2-nœuds:

$$N_1(\xi) = \frac{1}{2} . (1 - \xi)$$

$$N_2(\xi) = \frac{1}{2} . (1 + \xi)$$

Lagrange 3-nœuds:

$$N_1(\xi) = \frac{\xi}{2} . (\xi - 1)$$

$$N_2(\xi) = (\xi + 1) . (1 - \xi)$$

$$N_3(\xi) = \frac{\xi}{2} . (\xi + 1)$$

Lagrange 4-nœuds:

$$N_1(\xi) = \frac{9}{16} . (\xi - 1) . (\xi - \frac{1}{3}) . (\xi + \frac{1}{3})$$

$$N_2(\xi) = \frac{27}{16} . (\xi - 1) . (\xi + 1) . (\xi - \frac{1}{3})$$

$$N_3(\xi) = \frac{27}{16} . (\xi - 1) . (\xi + 1) . (\xi + \frac{1}{3})$$

$$N_4(\xi) = \frac{9}{16} . (\xi + 1) . (\xi - \frac{1}{3}) . (\xi + \frac{1}{3})$$

Eléments d'Hermite (continuité C^1) :

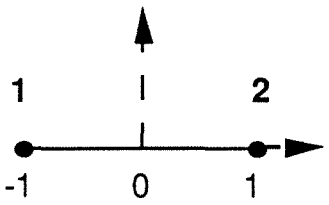


Figure AIV-2 : repère de référence de l'éléments d'Hermite 2-nœuds

Nous obtenons les fonctions de forme suivantes, exprimées dans le repère de référence :

Hermite 2-nœuds:

$$N_1(\xi) = (2 - 3.\xi + \xi^3)/4$$

$$N_2(\xi) = (1 - \xi - \xi^2 + \xi^3)/4$$

$$N_3(\xi) = (2 + 3.\xi - \xi^3)/4$$

$$N_4(\xi) = (-1 - \xi + \xi^2 + \xi^3)/4$$

Annexe V : Lissage par fonctions splines

Dans les bureaux d'études, il n'y a pas si longtemps, les dessinateurs utilisaient des baguettes de bois souple pour tracer des courbes passant par un ensemble de points donnés. La forme de la baguette déformée fournissait une courbe lisse correspondant à la minimisation de l'énergie de flexion. Une telle baguette était appelée "Spline".

Nous supposons que nous disposons de n points connus par leurs coordonnées (x_i,y_i) et que nous désirons tracer la courbe y=f(x) en utlisant un lissage par des fonctions splines cubiques . Pour cela, sur chaque intervalle $[x_{i-1}, x_i]$ i=1,...,n-1, nous allons approcher la courbe par une cubique en imposant aux bornes de l'intervalle une continuité sur la valeur y, sur la pente y' et sur la courbure y" .

Nous noterons $J_i = [x_i, x_{i+1}]$ et $h_i = x_{i+1} - x_i > 0$

Sur J_i, nous écrivons l'équation de la cubique par

$$y = a_i . (x - x_i)^3 + b_i . (x - x_i)^2 + c_i . (x - x_i) + d_i \qquad (1)$$

$$\text{en} \begin{cases} x = x_i & y = y_i \\ x = x_{i+1} & y = y_{i+1} \end{cases}$$

Ceci implique :

$$\begin{cases} y_i = d_i \\ y_{i+1} = a_i . h_i^3 + b_i . h_i^2 + c_i . h_i + d_i \end{cases} \qquad (2)$$

Les pentes et les courbures sont données respectivement par :

$$\begin{cases} y' = 3.a_i.(x-x_i)^2 + 2.b_i.(x-x_i) + c_i & (3) \\ y'' = 6.a_i.(x-x_i) + 2.b_i & (4) \end{cases}$$

Nous noterons par la suite $S_i = y''(x_i)$

Si nous imposons la continuité de la courbure, on obtient les realations suivantes :

$$\text{en } (x_i, y_i) \qquad S_i = 2.b_i \qquad (5a)$$

$$\text{en } (x_{i+1}, y_{i+1}) \qquad S_{i+1} = 6.a_i.h_i + 2.b_i \qquad (5b)$$

$$\text{soit} \quad a_i = \frac{(S_{i+1} - S_i)}{6.h_i} \quad \text{et} \quad b_i = \frac{S_i}{2} \qquad (5c)$$

En réintroduisant ces valeurs dans (2), on obtient sans difficulté :

$$c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{2.h_i.S_i + h_i.S_{i+1}}{6} \qquad (6)$$

Si l'on exprime maintenant la continuité de la tangente, on obtient :

$$y'_i = c_i = 3.a_{i-1}.h_{i-1}^2 + 2.b_{i-1}.h_{i-1} + c_{i-1} \qquad (7)$$

En utilisant les relations (2), (5), (6) et (7) écrites sur les intervalles J_i et j_{i-1}, on obtient finalement une relation ne faisant intervenir que les S_i :

$$h_{i-1}.S_{i-1} + 2.(h_{i-1} + h_i).S_{i-1} + h_i.S_i = 6.\left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right) \qquad (8)$$

Cette équation peut être écrite pour i=2,...,n-1 .

Nous remarquons que si nous sommes capables de déterminer les S_i, i=1,...,n, alors tous les coefficients a_i, b_i, c_i et d_i seront connus et donc toutes les cubiques cherchées . Les relations (8) nous donnent n-2 conditions, il en manque encore deux : sur S_1 et S_n .

Plusieurs solutions sont possibles :

- a) $S_1 = S_n = 0$ (splines naturels) . La courbure est nulle aux extrémités; la courbe s'approche linéairement de ses extrémités .
- b) $S_1 = S_2$; $S_{n-1} = S_n$: la courbe s'approche paraboliqueement de ses extrémités .
- c) extrapoler linéairement S_1 et S_n à partir de S_2-S_3 et de S_{n-2}-S_{n-1} respectivement .

Nous choisissons la condition a) . Dès lors, nous avons à résoudre un système tribande composé des équations suivantes :

$$\begin{cases} s_1 = 0 \\ h_{i-1} \cdot S_{i-1} + 2.(h_{i-1} + h_i) \cdot S_{i-1} + h_i \cdot S_i = 6. \left(\dfrac{y_{i+1} - y_i}{h_i} - \dfrac{y_i - y_{i-1}}{h_{i-1}} \right) \qquad i=2,..,n-1 \\ s_n = 0 \end{cases}$$

Ce système est résolu très aisément par pivot de Gauss . On en déduit alors les coefficients a_i, b_i, c_i et d_i puis les équations des cubiques sur chaque intervalle . On peut aussi en déduire la valeur de la dérivée par l'équation (3) . Ceci donne de bons résultats sauf aux extrémités de l'intervalle, points qui nécessitent un calcul à part (n'utilisant pas les splines) .

Une approche plus mathématique mais totalement équivalente à celle que nous venos de faire consiste à chercher une fonction S(x) minimisant:

$$J(x) = \int_{x1}^{xn} S(x)^2 \, . \, dx \qquad (9)$$

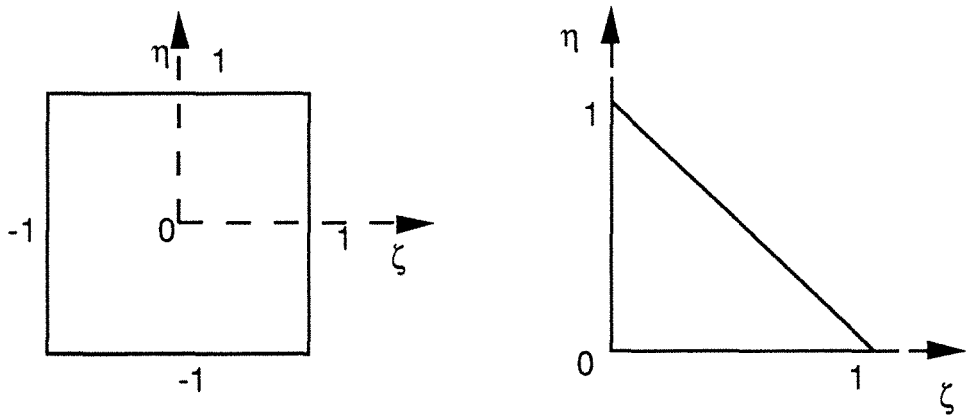
La fonction S(x) est évidemment la fonction courbure . On retrouve ici une approche plus mécanique (minimisation de l'énergie de flexion) . L'explicitation de ce minimum en utilisant les points imposés (xi,yi) conduit aux équations (8) .

Ce type d'approche est plus général, il permet de définir des lissage par splines pondérés [GR 6] offrant une plus grande stabilité dans le tracé .

L'ensemble du processus donne déjà de bons résultats et est utilisé par nos logiciels .

Annexe VI : Fonctions de forme des élément utilisés en élasticité plane

les fonctions de forme de chaque élément sont exprimées dans le repère de référence de cet élément correspondant à la figure AVI-1 ci-dessous . Les nœuds sont tous numérotés dans le sens trigonométrique à partir du nœud inférieur gauche (de coordonnées (-1,-1) pour les quadrilatères et (0,0) pour les triangles) . Dans le cas du quadrilatère 9-nœuds, le neuvième nœud est le nœud central.



élément de référence quadrilatère (4,8 ou 9 nœuds)

élément de référence triangulaire (3 ou 6 nœuds)

Figure AVI-1 : Repères de référence des éléments utilisés

Triangle à 3 nœuds

$$N_1(\xi,\eta) = 1 - \xi - \eta$$

$$N_2(\xi,\eta) = \xi$$

$$N_3(\xi,\eta) = \eta$$

Quadrilatère à 4 nœuds

$$N_1(\xi,\eta) = \frac{1}{4}.\,(1-\xi)\,.\,(1-\eta)$$

$$N_2(\xi,\eta) = \frac{1}{4}.\,(1+\xi)\,.\,(1-\eta)$$

$$N_3(\xi,\eta) = \frac{1}{4}.\,(1+\xi)\,.\,(1+\eta)$$

$$N_4(\xi,\eta) = \frac{1}{4}.\,(1-\xi)\,.\,(1+\eta)$$

Triangle à 6 nœuds

$$N_1(\xi,\eta) = 1 - 3.\xi - 3.\eta + 2.\xi^2 + 2.\eta^2 + 4.\xi.\eta$$

$$N_2(\xi,\eta) = 4.\xi - 4.\xi^2 - 4.\xi.\eta$$

$$N_3(\xi,\eta) = -\xi + 2.\xi^2$$

$$N_4(\xi,\eta) = 4.\xi.\eta$$

$$N_5(\xi,\eta) = -\eta + 2.\eta^2$$

$$N_6(\xi,\eta) = 4.\eta - 4.\eta^2 - 4.\xi.\eta$$

Quadrilatère à 8 nœuds

$$N_1(\xi,\eta) = \frac{1}{4}.(1-\xi).(1-\eta).(-1-\eta-\xi)$$

$$N_2(\xi,\eta) = \frac{1}{2}.\left(1-\xi^2\right).(1-\eta)$$

$$N_3(\xi,\eta) = \frac{1}{4}.(1+\xi).(1-\eta).(-1+\eta-\xi)$$

$$N_4(\xi,\eta) = \frac{1}{2}.\left(1-\eta^2\right).(1+\xi)$$

$$N_5(\xi,\eta) = \frac{1}{4}.(1+\xi).(1+\eta).(-1+\eta+\xi)$$

$$N_6(\xi,\eta) = \frac{1}{2}.\left(1-\xi^2\right).(1+\eta)$$

$$N_7(\xi,\eta) = \frac{1}{4}.(1-\xi).(1+\eta).(-1+\eta-\xi)$$

$$N_8(\xi,\eta) = \frac{1}{2}.\left(1-\eta^2\right).(1-\xi)$$

Quadrilatère à 9 nœuds

$$N_1(\xi,\eta) = \frac{1}{4}.(\xi^2-\xi).(\eta^2-\eta)$$

$$N_2(\xi,\eta) = \frac{1}{2}.(1-\xi^2).\eta.(\eta-1)$$

$$N_3(\xi,\eta) = \frac{1}{4}.(\xi^2+\xi).(\eta^2-\eta)$$

$$N_4(\xi,\eta) = \frac{1}{2}.(1-\eta^2).\xi.(\xi+1)$$

$$N_5(\xi,\eta) = \frac{1}{4}.(\xi^2+\xi).(\eta^2+\eta)$$

$$N_6(\xi,\eta) = \frac{1}{2}.(1-\xi^2).\eta.(\eta+1)$$

$$N_7(\xi,\eta) = \frac{1}{4}.(\xi^2-\xi).(\eta^2+\eta)$$

$$N_8(\xi,\eta) = \frac{1}{2}.(1-\eta^2).\xi.(\xi-1)$$

$$N_9(\xi,\eta) = (1-\eta^2).(1-\xi^2)$$

Références bibliographiques

Nous avons organisé nos références bibliographiques par thèmes, chacun représenté par deux lettres majuscules. Ces deux lettres suivis d'un numéro, constitueront le point commun de tous les références traitant du même thème. Les références polyvalentes seront classées selon le thème majoritaire apparaissant .

Les thèmes sont les suivants :

[GR] : traite des problèmes de graphisme, de représentation et de calcul des parties cachées .

[EL] : traite de la modélisation générale par éléments finis .

[MG] : traite de la théorie des méthodes multigrilles .

[AN] : traite de la théorie et de la pratique générale de l'analyse numérique .

[IN] : rassemble tous les articles publiés par l'INRIA (Institut National de Recherche en Informatique et en Automatique) . Les domaines couverts sont nombreux et variés; nous nous sommes cependant particulièrement intéressés aux articles traitant des problèmes de mise en œuvre de la méthode des éléments finis.

[ME] : traite des problèmes généraux de mécanique des solides.

[IF] : traite des problèmes de programmation soit au niveau général (langages, structures etc...) ou bien au niveau particulier de l'utilisation du Mac Intosh.

Liste des références

Références sur les éléments finis

[EL 1] : **O.C Zienkiewicz et R.L Taylor :** "The finite element method". 4th Ed . McGraw-Hill . (Tome I 1987 , tome II 1990).

[EL 2] : **G Dhatt et D Touzot** : "Une présentation de la méthode des éléments finis" . Collection UTC , Editeur Maloine (1981).

[EL 3] : **E Hinton et D.R.J Owen** : "An introduction to finite element computations" . Pineridge Press Limited. (1979)

[EL 4] : **D.R.J Owen et E Hinton** : "Finite element in plasticity : theorie and practice" . Pineridge Press Limited. (1980)

[EL 5] : **E Hinton et D.R.J Owen** : "Finite element software for plates and shells" . Pineridge Press Limited. (1984)

[EL 6] : **J.L Batoz et G Dhatt** : "Modélisation des structures par éléments finis . Volume 1 : Solides élastiques" . Hermès . (1990).

[EL 7] : **J.L Batoz et G Dhatt** : "Modélisation des structures par éléments finis . Volume 2 : Poutres et plaques" . Hermès. (1990) .

[EL 8] : **M Bernardou et Al** : "MODULEF : Une bibliothèque modulaire d'éléments finis" . INRIA. (1988).

[EL 9] : **P.L George** : "MODULEF : Génération automatique de maillages". Collection didactique INRIA. (1987).

[EL 10] : **K.J Bathe** : "finite element procedures in engineering analysis". Prentice-Hall INc . (1982).

[EL 11] : **R.H Gallagher** : "Introduction aux elements finis" . Pluralis. (1976).

[EL 12] : **O Pironneau** : "Méthodes des éléments finis pour les fluides". Masson . (1988).

[EL 13] : **C Taylor et T.J Hughes** : "Finite element programming of the Navier-Stokes equation". Pineridge Press Limited. (1980).

[EL 14] : **J.C Sabonnadière** : "Eléments finis et CAO" . Hermès . (1986).

[EL 15] : **D.R.J Owen et A.J Fawkes** : "Fracture mechanics" . Pineridge Press Limited. (1983).

[EL 16] : **P.L George** : "Génération automatique de maillages : Application aux méthodes d'éléments finis" . Masson. (1991).

[EL 17] : **M Gueury** : "Cours d'éléments finis avancés" . ENSEM-DEA de mécanique énergétique.

[EL 18] : **A.M Baudron et P Trompette** : "Un critère de jugement simple pour un maillage éléments finis" . Journal de mécanique théorique et appliquée. Vol 5, N° 5. (1986).

[EL 19] : **O Pironneau** : Notice explicative du logiciel MacFEM . Numerica.

Références sur les problèmes de graphisme

[GR 1] : **T Liebling et H Röthlisberger** : "Infographie et applications" . Masson. (1988).

[GR 2] : **R Dony** : "Calcul des parties cachées" . Masson . (1986).

[GR 3] : **J Woodmark** : "Calcul de formes par ordinateur" . Masson . (1988).

[GR 4] : **D.F Rogers** : "Algorithmes pour l'infographie" . McGraw-Hill. (1988).

[GR 5] : **R Dony** : "Graphisme scientifique sur ordinateur : de la 2^e à la 3^e dimension". Masson. (1985).

[GR 6] : **Y Gardan** : "La CFAO" . Hermès . (1986).

[GR 7] : **P Lancaster et K Salkaukas** : "Curve and surface fitting". Academic Press. (1986).

Références sur les méthodes multigrilles

[MG 1] : **W Hackbusch** : "Multi-grid methods and applications" . Springer-Verlag . (1985) .

[MG 2] : **W Hackbusch et U Trottenberg** : "multigrid methods : proceedings, Köln-Porz, 1981" . Springer-Verlag . (1982) .

[MG 3] : **W.L Briggs** : "A multigrid tutorial" . SIAM . (1897) .

[MG 4] : **J.F Maitre** : "Cours d'introduction aux méthodes multigrilles" . (1985) .

[MG 5] : **F Pirali** : "Méthodes multigrilles" . Rapport de DEA . (Nancy 1986) .

[MG 6] : **C Le Carlier De Veslud** : "Application des méthodes multigrilles à la mécanique des structures" . Rapport de DEA . (Nancy 1987) .

[MG 7] : **C Le Carlier De Veslud, G Maurice et R Kouitat** : "A multigrid mocrosoftware for engineers on Macintosh computer" . Publication dans "Computational Mechanics Software for Engineering Workstations" . 1992

[MG 8] : **R.A Nicolaïdes** : "On multiple grid and related techniques for solving discrete elliptics systems" . Journal of computational physics 19 . (1975) .

[MG 9] : **C Auburtin** : "Analyse de structures poutres planes linéaires et non-linéarité par éléments finis et méthodes multigrilles sur micro-ordinateur" . Thèse de l'I.N.P.L. 1990.

[MG 10] : **C Auburtin** : "Analyse de structures poutres planes en non-linéarité géométrique par éléments finis et méthodes multigrilles sur micro-ordinateur" . Communication au 10ème Congrès Français de Mécanique . Paris. 1991.

Publications de l'INRIA

[IN 1] : **P.L George et F Hermeline** : "Maillage de Delaunay d'un polyèdre convexe en dimension d . Extension à un polyèdre quelconque" . Rapport de recherche N° 967 . INRIA . (1989) .

[IN 2] : **P.L George, F Hecht et E Saltel** : "Tétraédrisation automatique et respect de la frontière" . Rapport de recherche N° 835 . INRIA . (1988) .

[IN 3] : **F Benkhaldoun, T Fernandez, B Larrouturou et P Leyland** : "A dynamical adaptative method based on local reffinement and unrefinement for triangular finite-element meshes : preliminary results" . Rapport de recherche N° 1271 . INRIA . (1990)

[IN 4] : **M.H Lallemand, H Steve et A Dervieux** : "Unstructured multigridding by volume aglomeration : current status" . Rapport de recherche N° 1224 . INRIA . (1990)

[IN 5] : "Huitième colloque international sur les méthodes de calcul scientifiques et technique" . Edition provisoire . Particulièrement le tome II, pages 605 à 665 . INRIA . (1987) .

[IN 6] : **P.L George et A Golgolab** : "Mailleur 3D en topologie "cylindrique" " . Rapport de recherche N° 100 . INRIA . (1988)

[IN 7] : **B Muller** : "Les outils graphiques de la bibliothèque MODULEF" . Rapport de recherche N° 98 . INRIA . (1988) .

[IN 8] : **M Bernardou** : "Formulation variationnelle, approximation et implémentation de problèmes de barres et de poutres bi- et tri-dimensionnelles . Partie A : Barres et poutres tridimensionnelles" . Rapport de recherche N° 731 . INRIA . (1987) .

[IN 9] : **M Bernardou** : "Formulation variationnelle, approximation et implémentation de problèmes de barres et de poutres bi- et tri-dimensionnelles . Partie B : Barres et poutres bidimensionnelles" . Rapport de recherche N° 87 . INRIA . (1987) .

[IN 10] : **P.L George** : "Mailleur 3D par découpage structure d'éléments grossiers" . Rapport de recherche N° 990 . INRIA . (1989) .

[IN 11] : **A Golgolab** : "Mailleur 3D automatique pour des géométries complexes" . Rapport de recherche N° 1004 . INRIA . (1989) .

[IN 12] : **F Hecht et E Saltel** : "EMC^2 un logiciel d'édition de maillages et de contours bidimensionnels" . Rapport de recherche N° 118 . INRIA . (1990) .

[IN 13] : **S Gopalsamy et O Pironneau** : "Interpolation C^1 de résultats C^0 " . Rapport de recherche N° 100 . INRIA . (1989)

[IN 14] : **M.H Lallemand, F Fezoui et E Perez** : "Un schéma multigrilles en éléments finis décentré pour les équations d'Euler" . Rapport de recherche N° 602 . INRIA . (1987) .

[IN 15] : **F Angrand et P Leyland** : "Schéma multigrille dynamique pour la simulation d'écoulements de fluides visqueux compressibles" . Rapport de recherche N° 659 . INRIA . (1987) .

[IN 16] : **B Mercier et O Pironneau** : "Some exemples of implementation and of application of the finite element method" . Rapport de recherche N° 248 . INRIA . (1977) .

[IN 17] : **P.L George et J.L Nizard** : "Maillage aigu" . Rapport de recherche N° 1414 . INRIA . (1991) .

[IN 18] : **B Palmerio :** "Self-adaptative FEM algorithms for the Euler equations" . Rapport de recherche N° 338 . INRIA . (1984) .

[IN 19] : **E Perez** : "Finite element and multigrid solution of the two dimensionnal Euler equations on a non-structured grid" . Rapport de recherche N° 442 . INRIA . (1985) .

Références d'analyse numérique

[AN 1] : **G.H Golub et G.A Meurant** : "Résolution numérique des grands systèmes linéaires" . Eyroles . (1983) .

[AN 2] : **R Dautray et J.L Lions** : "Analyse numérique et calcul numérique" . Masson (1988) .

[AN 3] : **G Maurice** : "Cours d'analyse numérique" . ENSEM .

[AN 4] : **A George et W Liu** : "computer solution of large sparse positive definite systems" . Prentice-Hall . (1981) .

[AN 5] : **R Glowinski et J.L Lions** : "Analyse numérique des inéquations variationnelles" . Tome 1 . Dunod . (1976) .

Références de mécanique

[ME 1] : **H Ducauquis** : "Cours de mécanique des milieux continus" . ENSEM-DEA de mécanique énergétique .

[ME 2] : **I Popov** : "Introduction to mecanics of solid" . Prentice-Hall . (1968) .

[ME 3] : **W.C Young** : "Roark's formulas for stress and strain" . McGraw-Hill . (1989) .

[ME 4] : **P Germain** : "Mécanique" . Ellipses. (1986) .

[ME 5] : **Y Bamberger** : "Mécanique de l'ingénieur . Tome II : Milieux déformables" . Herman. (1981) .

Références d'informatique

[IF 1] : "Inside Macintosh" . Tomes I à V .Addison-Wesley Publishing Company Inc . (1985) .

[IF 2] : "Microsoft FORTRAN" . Manuel d'utilisation version Macintosh.

[IF 3] : **P Le Beux** : "Introduction à Mac Basic" . Sybex . (1986) .

INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

INPL

AUTORISATION DE SOUTENANCE DE THESE
DU DOCTORAT DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

VU LES RAPPORTS ETABLIS PAR :
Monsieur PIERRE, Professeur, Université de Nancy I,
Monsieur PIRONNEAU, Professeur, INRIA Rocquencourt.

Le Président de l'Institut National Polytechnique de Lorraine, autorise :

Monsieur LE CARLIER DE VESLUD Christian

à soutenir devant l'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE, une thèse intitulée :
"Etude et réalisation de logiciels d'éléments finis sur micro-ordinateur. Mailleurs automatiques. Résolution auto-adaptative de problèmes de mécanique utilisant la méthode multigrilles"

en vue de l'obtention du titre de :

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

Spécialité : "MECANIQUE ET ENERGETIQUE"

Fait à Vandoeuvre le, **24 Octobre 1991**
Le Président de l'I.N.P.L.,
M. GANTOIS

2, avenue de la Forêt de Haye - B.P. 3 - 54501 VANDŒUVRE CEDEX
Téléphone : 83. 59. 59. 59 - Télex : 961 715 F - Télécopie : 83. 59. 59. 55

Résumé

Le travail présenté ici consiste en l'étude et le développement d'un ensemble de logiciels sur micro-ordinateur Macintosh pour la simulation numérique de problèmes de mécanique utilisant les méthodes multigrilles. On peut y distinguer deux parties:

- la première consacrée aux méthodes de maillage automatique
- la seconde traitant des applications mécaniques: la simulation de problèmes d'ossatures planes linéaires de poutres et d'élasticité linéaire plane.

Les difficultés essentielles du projet consistent, d'une part à s'adapter aux possibilités limitées du micro-ordinateur pour traiter des problèmes qui peuvent être très vite de taille importante, et d'autre part, de garantir la fiabilité du logiciel tout en offrant à l'utilisateur simplicité d'emploi et automaticité utilisant au maximum la convivialité de l'appareil.

La première partie concerne la réalisation de logiciels de maillage automatiques en 2D et 3D. Deux logiciels sont présentés dans chaque cas. Un premier choix offre un maillage par blocs grossiers 2D ou 3D définis par une transformation, puis maillés séparément couche par couche. Un deuxième choix 2D est la méthode de Delaunay qui permet de construire automatiquement une triangulation à partir du contour. Enfin, un mailleur 3D par élévation cylindrique de maillages 2D est mis en place. Divers utilitaires y sont adjoints: modifications par suppressions ou adjonctions d'entités, reproduction par transformations géométriques simples, visualisation et calcul des parties cachées...etc.

La seconde partie est consacrée au développement systématique des méthodes multigrilles sur micro-ordinateur. Nous présentons une structure de données spécifique associée à la méthode Full-multigrilles ainsi que des processus de raffinement automatique. Les outils précédemment développés sont utilisés à deux types d'applications. D'abord, le calcul de la déformation d'ossatures planes linéaires de poutres avec ou sans cisaillement transverse. Enfin, une application à l'élasticité plane bidimensionnelle est présentée. Cette dernière est complétée par un processus de raffinement auto-adaptatif du maillage tenant compte des erreurs locales. Ce qui permet au logiciel de concentrer ses efforts là où c'est nécessaire, économisant ainsi place mémoire et temps de calcul, une optique particulièrement bien adaptée au cas des micro-ordinateurs.