RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

Université Dr. Yahia Farès de Médéa
Faculté des Sciences et de Technologie



Mémoire de Magister

Spécialité : *Génie Mécanique*
Option : *Énergétique Numérique*

Présenté par
MENAD Ahmed

Thème :

Elaboration d'un générateur de maillage : application aux éléments finis quadratiques 2D

A soutenir le 21 /07/2011 devant le jury composé de :

Pr ABUDURA Salam	**Professeur**	**U.Y.F.Medea**	**Président**
Dr Abdellah El Hadj Abdellah	**Maître de Conférences**	**U.Y.F.Medea**	**Examinateur**
Dr BOUAZIZ Mohamed Nadjib	**Maître de Conférences**	**U.Y.F.Medea**	**Examinateur**
Dr Amar BOUZID Djillali	**Maître de Conférences**	**U.Y.F.Medea**	**Directeur de Mémoire**
Dr ALLICHE Mounir	**Maître de Conférences**	**U.Y.F.Medea**	**Invité**

2010/2011

Dédicaces

Au-delà des personnes, des lieux et des temps, je dédie ce travail avec ma profonde conviction, à tous ceux qui ont toujours cru et sans doute à la Science. Cette lumière qui éclaire les esprits et leur permet de transcender les limites installées par les sociétés et les cultures dans leurs périodes de dégénérescence.

À ceux qui savent concrétiser leurs idées et savent réussir.
À ceux qui ne portent que du bien pour les autres.
À ceux qui ne vivent que dans et par la vérité.

Je dédie ce travail à :
Toute ma famille (sara, amina, abd el kader, merouan, bilal…), et en particulier à mes parents et mon frère, pour leur soutien indéfectible durant ces années d'étude. Je suis heureux d'avoir pu lire de la joie et de la fierté dans leur regard.
Enfin j'exprime ma reconnaissance et ma sympathie à, mes collègues de magister, mes frères dans la foi en particulier Mourad Bouzair, Farouk Ziani, Youcef Boudaoud,…etc. à toute personne qui de prés ou de loin a rendu agréable mon séjour à Médéa.

MENAD AHMED

REMERCIEMENTS

Je me dois tout d'abord remercier Allah le Tout Puissant pour toute la volonté et le courage qu'il m'a donné pour l'achèvement de ce travail.

Je tiens avant tout à exprimer ma très grand gratitude à Monsieur **Dr. Amar Bouzid Djillali,** Maître Conférence à l'Université de Médéa, non seulement pour m'avoir proposé le sujet et accepté d'encadrer ce travail, mais surtout pour m'avoir insufflé le désir et la passion de la recherche, qu'il trouve dans ces mots l'expression de mes vifs remerciements.

Je tiens à exprimer ma profonde reconnaissance à Monsieur **ABUDURA Salam** Professeur à l'Université de Médéa pour m'avoir fait l'honneur de présider le jury de mon mémoire de magister.

J'adresse mes sincères remerciements au Dr **Abdellah El Hadj Abdellah** et Dr **BOUAZIZ Mohamed Nadjib** Maîtres de Conférences à l'Université de Médéa pour leur participation à mon jury de mémoire et examiner ce travail.

Mes remerciements amicaux s'adressent à Messieurs Dr. **Alliche Mounir** Maître de Conférence à l'Université de Médéa, pour leur aide sans limite dans la programmation, pour leur encouragement et leurs orientations durant l'élaboration de ce travail.

Résumé :

Dans ce travail, on a abordé le développement d'un pré-processeur MESH_2D en FORTRAN 77 pour la tâche de génération automatique de maillage, se base sur la méthode par blocs, On a adopté une stratégie qui consiste à éliminer la répétition des nœuds, avec un système de numérotation automatique des nœuds inclus et bien choisi permettant de minimiser la connectivité nodale du maillage généré et minimiser la largeur de bande des matrices, ce qui permet aux éléments de différents types (linéaires ou quadratiques) d'être produits simultanément lors de la génération de maillage.

Pour cela, on a adapté les fichiers des résultats du code mailleur MESH_2D au fichier d'entrée de code solveur (MAT-fem) utilisant l'analyse par la méthode des éléments finis pour des applications dans le domaine solide et thermique.

Mots clés: pré-processeur, méthode par blocs, éléments (linéaires ou quadratiques), génération de maillage, connectivité, la méthode des éléments finis, solveur (MAT-fem).

Abstract :

In this work, we exposed the development of a preprocessor (MESH_2D) in FORTRAN 77 for the task of automatic mesh generation, based on the method block, We adopted a strategy of eliminating duplication of nodes, with an included and well chosen system of automatic numbering of nodes to minimize the nodal connectivity of the mesh generated and minimize the bandwidth of matrices, which allows elements of different types (linear or quadratic) to be generated simultaneously at mesh generation.

For this, we adapted the results files of (MESH_2D) mesh generator code to input code solver file (MAT-fem), using the finite element method for applications in solid and heat analysis.
Key words : preprocessor, method block, elements (linear or quadratic), mesh generation, connectivity, finite element method, solver (MAT-fem).

ملخص :

في هذا العمل ، عرضنا تطوير برنامج قبل المعالجة (MESH_2D) بلغة فورتران 77 لمهمة توليد الشبكي الآلي ، إعتماداً على طريقة الكتل ، تبنينا إستراتيجية إزالة العقد مضاعفة ، مع الاختيار الجيد لنظام ترقيم الآلي للعقد الذي يسمح بتقليل من ربط العقدي لشبكات مولدة وتقليص عرض الشريط للمصفوفات ، هذا يسمح لكافة عناصر من أنواع مختلفة (خطية أو تربيعية) لكي تولد في نفس الوقت في مولد شبكي.

لهذا, كيفنا ملف نتائج (MESH_2D) مع ملف معطيات لحلال رقمي(MAT-fem) وذلك بإستعمال طريقة العناصر المنتهية من أجل تطبيقات في مجال حراري وصلب.

كلمات المفتاح : برنامج قبل المعالجة , طريقة الكتل, عناصر (خطية أو تربيعية), مولد شبكي, ربط العقدي, طريقة العناصر المنتهية, حلال رقمي.

TABLE DE MATIERES

Page

Résumé …… *i*
Dédicace……. *ii*
Remerciements…….. *iii*
Table de matières……... *iv*
Nomenclature et symboles…….. *v*
Liste des figures……... *vi*
Liste des tableaux……. *vii*

Introduction
1. Introduction générale……... 1
2. Problème de la génération de maillage automatique…….. 3
3. Contenu du mémoire……... 3

Chapitre 1 : Synthèse bibliographique
1.1 Historique sur la génération de maillage……... 5
1.2 Travaux réalisés dans le domaine de la génération de maillage automatique 6

Chapitre 2 : Méthodes de génération de maillage
2. Introduction……. 11
2.1 Les maillages structurés……. 11
2.1.1 Méthodes algébriques…… 12
2.1.1.1 Introduction…….. 12
2.1.1.2 Transformations conformes…… 15
2.1.1.2.1 La transformation $\omega = z^2$ 17
2.1.1.2.2 La transformation $\omega = e^z + z$……. 18
2.1.2 Méthodes de blocs…….. 22
2.1.3 Méthodes par résolution d'EDP …….. 23
2.1.4Autres méthodes ……. 23
2.2 Les maillages non structurés…… 24
2.2.1 Méthode de type Delaunay-Voronoï…… 25
2.2.1.1 Triangulation de Delaunay…….. 25
2.2.1.2 Mailleur de Delaunay……... 27
2.2.1.3 Critique de la méthode…… 29
2.2.2 Méthodes de type frontales…….. 30
2.2.2.1 Description de la méthode……... 30
2.2.2.2 Algorithme……. 32
2.2.2.3 Un meilleur frontal…… 32
2.2.2.4 Le front initial……. 33
2.2.2.5 L'analyse du front ……. 33
2.2.2.6 Création du sommet……. 35
2.2.2.7 Position de sommet cas1……... 35
2.2.2.8 Position de sommet cas2……... 36
2.2.2.9 Coût d'un meilleur frontal…… 36

2.2.2.10 Optimisation d'un mailleur frontal……... 37
2.2.2.11 Critique de la méthode……... 37
2.2.3 Méthodes de quadtree-octree (méthode par arbre)…… 38
2.2.3.1 Critique de la technique……... 39
2.2.4 Autres méthodes…… 39
2.2.4.1 Méthode de « paving »…… 40
2.2.4.2 Méthode de squelette…….. 40
2.2.4.3 Méthode topologique…….. 41
2.3 Quelques problèmes fréquents……... 43

Chapitre 3 : Méthodes de blocs

3.1 Méthode de blocs en dimension 2……... 44
3.1.1 Algorithme de maillage…….. 44
3.1.2 Mailleur à super éléments…….. 47

Chapitre 4 : Présentation du programme développé

4. Programme informatique……... 50
4.1 Introduction…… 50
4.2 Génération d'un maillage automatique…….. 51
4.2.1 Division de la structure en blocs…… 51
4.2.2 Subdivision de chaque bloc en éléments…….. 52
4.2.3 Connections des blocs……..... 52
4.3 Description du programme MESH_2D…….. 53
4.3.1 Introduction des données et sortie des résultats……. 53
4.2.3.1 Description le fichier de données……. 53
4.3.2 Description les sous programmes de mailleur (MESH_2D)……. 55
4.3.2.1 Le sous-programme INPUT……. 56
4.3.2.2 Le sous-programme REORDR……. 56
4.3.2.3 Le sous-programme MESH…….. 58
4.3.2.4 Le sous-programme LINEAR…….. 59
4.3.2.5 Le sous-programme QUAD…… 61
4.3.2.6 Le sous-programme CORD……. 63
4.3.2.7 Sous-programme RENMBR, OPTNUM……..... 64
4.3.2.8 Sous-programme SORTIE…… 64
4.3.3 Notations du programme (MESH_2D)…….. 64
4.4 Description du programme **MAT-fem…….** 66
4.4.1 Début……... 69
4.4.2 Matrice de rigidité et vecteur de force nodal équivalent……..... 70
4.4.2.1 Généralités…… 70
4.4.2.2 Calcul et assemblage de **K (e)** et **f (e)** (poids propre) pour les triangles à 3 nœuds et des quadrilatères à 4 nœuds……. 72
4.4.3 Charges externes……..... 73
4.4.4 Solutions du système d'équations…… 73
4.4.5 Calcule les réactions nodales……..... 74
4.4.6 Calcul les contraintes au niveau des nœuds…….. 77

4.4.7 Etape de post-traitement…… 75
4.4.8 Description Gid…… 76
4.4.9 Limites de calcul actuelles du générateur de maillage……..... 76

5. 4 Propriétés essentielles d'un mailleur 77

Chapitre 5 : Application

5. Application …….. 79
5. 1 Mise en marche du code et comparaison avec les résultats des travaux [3,15].. 79

5. 2 L'effet des types d'éléments…….. 82
5. 3 Application à des géométries complexes et l'effet de raffinement de maillage…… 93
5. 5 Application à des problèmes d'élasticités planes…… 104
5. 5.1 Introduction……... 104
5. 5.1.1 Etude de la flexion d'une poutre en acier encastrée soumise à une charge répartie constante…….. 104
5. 5.1.2 Etude de la flexion d'une poutre supportée aux extrémités sous l'effet d'une charge……. 109
5. 5.1.3 Etude de la flexion d'une poutre en acier encastrée soumise à une charge constante……. 113
5. 6 Application à des problèmes thermiques……..... 118
5. 6.1 Introduction…….. 118
5. 6.1.1 conductions thermiques dans une plaque……. 118
5. 6.1.2 modélisations thermiques de four…….. 131

Conclusion générale… … 146
Références bibliographiques……... 148
Annexes 1…… 153
Annexes 2…… 171
Annexes 1…… 174

Nomenclature et symboles

Symbole	Quantité	Unité de mesure
	Chapitre 2	
$u = (u^1, u^2, u^3)$	les coordonnées dans l'espace paramétrique	m
$x=(x_1,x_2,x_3)$	coordonnées dans l'espace physique	m
N,S	ensemble de nœuds	
Z, ω	fonction d'une variable complexe	
Ω	la frontière de maillage	
	Chapitre 3	
N_i	fonction de forme	
nb-nœuds	le nombre de nœuds de l'élément	
ξ et η	les coordonnées d'un point de l'élément de référence	m
$y(\xi,\eta)$ et $x(\xi,\eta)$	les coordonnées d'un point de l'élément réel	m
x_i et y_i	les coordonnées du *i*e nœud de l'élément	m
	Chapitre 4	
npnod	nombre de nœuds	
nndof	nombre total de degrés de liberté	
nelem	nombre d'éléments	
nnode	nombre de nœuds par élément	
neleq	nombre de degrés de liberté par élément	
neleq	Nombre de degrés de liberté par élément	
[K]	matrice caractérisant le système	
{U}	variables inconnues du problème	
{F}	sollicitations connues (second membre)	
ctder	Le vecteur ctder contient Les valeurs des dérivées cartésiennes des fonctions de forme à chaque point d'intégration	
lcffm	Le vecteur lcffm contient les valeurs des fonctions de forme évaluées au point d'intégration (i, j)	
lcder	la matrice lcder stocke les valeurs pour les dérivées naturelles des fonctions de forme à chaque point d'intégration (i, j).	
xcjacm	La matrice J	
TrStif et QdStif	la matrice de rigidité d'élément	
darea	Le différentiel de surface	
M	la matrice M	
Fy	Le vecteur de force nodal équivalent est stocké dans la variable Fy	
Gid	est un logiciel conçu dans le but de couvrir toutes les applications liées au domaine de la simulation numérique	
	Chapitre 5	
A	aire de la section droite	mm^2
E	module de YOUNG	MPa
ν	coefficient de Poisson	------
e	Epaisseur	mm

G	Module de cisaillement	N/mm²
H	Hauteur	mm
I_Z	moment quadratique suivant z	mm^4
L	Longueur	mm
P,F	une charge linéique	N/mm
λ	La conductivité thermique	W/m².°C
T	La température	°C
y_{max}	La flèche maximale de la poutre	mm

Liste des figures

Chapitre 1

Titre ***page***

Fig. 1.1 Plan de classification pour méthodes de génération de maillage. 07

Chapitre 2

Fig. 2.1 Transformation d'un maillage de l'espace paramétrique vers l'espace physique 13
Fig. 2.2 Transformation des frontières. 14
Fig. 2.3 Exemple d'une transformation algébrique 14
Fig. 2.4 Préservation des angles par les fonctions harmoniques. 16
Fig. 2.5 La transformation $\omega=z^2$. 17
Fig. 2.6 La transformation $\omega = e^z + z$ 18
Fig. 2.7 Maillages algébriques, sur un domaine «quadrangulaire » (à gauche) et « triangulaire » (à droite). 21
Fig 2.8 Maillage grossier et maillage final. 22
Fig. 2.9 Exemple de maillage par une méthode elliptique. 23
Fig. 2.10 Diagramme de Voronoï d'un nuage de nœuds et triangulation de Delaunay correspondante en 2D. 26
Fig. 2.11 méthode de type Delaunay en 2D, triangulation de l'enveloppe convexe (a) et reconstruction de la frontière (b). 28
Fig. 2.12 a) Quelque étape de front. 31
Fig. 2.12 b) Quelque étape de front (suite). 31
Fig. 2.13 Quelque étape de front (fin). 32
Fig. 2.14 Représente la discrétisation des frontières de front. 33
Fig. 2.15 Cas A) et construction associée 34
Fig. 2.16 Cas B) et construction associée 34
Fig. 2.17 Cas C) et construction associée 35
Fig. 2.18 les Deux constructions possibles pour la création du sommet 35
Fig. 2.19 Méthode par arbre en 2D. 39
Fig. 2.20 Maillage par la méthode « du tonnelier » 40
Fig. 2.21 Maillage d'un rectangle par une méthode de squelette. 41
Fig. 2.22 Maillage par une méthode topologique. 42
Fig. 2.23 Maillage par une méthode topologique (suite et fin). 42

Chapitre 3

Fig. 3.1: Numérotation formelle. 46
Fig. 3.2: Plaque trouée. 47
Fig. 3.3: a) l'objet à mailler b) décomposition de l'objet en 10 superéléments 48
Fig. 3.4: a) super-élément dans l'espace x et y. b) transformation du superélément dans l'espace de référence ξ et η et le maillage généré. c) transformation du maillage en l'espace x et y. 48

Chapitre 4

Fig. 4.1: élément quadratique isoparamétrique (8 nœud) 52
Fig. 4.2: représente l'organigramme du programme général 55
Fig. 4.3: indique l'organigramme du sous-programme REORDR 57
Fig. 4.4: représente l'organigramme du sous-programme MESH 58
Fig. 4.5: représente l'organigramme du sous-programme LINEAR 60
Fig. 4.6: représente l'organigramme du sous-programme QUAD 62
Fig. 4.7: représente l'organigramme du sous-programme CORD 63
Fig. 4.8: indique sous programme du fichier de données. 66
Fig. 4.9: représente sous programme plot2D 67
Fig. 4.10: indique L'organigramme du programme MAT-fem 68

Chapitre 5

Fig. 5.1: fichier de résultats de réf [3] 80
Fig. 5.2: fichier de résultats obtenus par notre code mesh_2D 80
Fig. 5.3: fichier de fichier de données de réf [3] 81
Fig. 5.4: Maillage généré par l'élément linéaire à trois nœuds 83
Fig. 5.5: Maillage généré par l'élément linéaire à 4 nœuds(1) 83
Fig. 5.6: Maillage généré par l'élément quadratique à 6 nœuds 84
Fig. 5.7: Maillage généré par l'élément quadratique à 8 nœuds 84
Fig. 5.8: Maillage généré par l'élément linéaire à trois nœuds 85
Fig. 5.9: Maillage généré par l'élément linéaire à 4 nœuds 86
Fig. 5.10: Maillage généré par l'élément quadratique à 6 nœuds 86
Fig. 5.11: Maillage généré par l'élément quadratique à 8 nœuds 87
Fig. 5.12: Maillage généré par l'élément linéaire à trois nœuds 88
Fig. 5.13: Maillage généré par l'élément linéaire à 4 nœuds(2) 88
Fig. 5.14: Maillage généré par l'élément quadratique à 6 nœuds 89
Fig. 5.15: Maillage généré par l'élément quadratique à 8 nœuds 89
Fig. 5.16: Maillage généré par l'élément linéaire à trois nœuds 90
Fig. 5.17: Maillage généré par l'élément linéaire à 4 nœuds 91

Fig. 5.18: Maillage généré par l'élément quadratique à 6 nœuds 91
Fig. 5.19: Maillage généré par l'élément quadratique à 8 nœuds 92
Fig. 5.20: Maillage généré par l'élément linéaire à 3 nœuds (N= 1260, E= 2352) 94
Fig. 5.21: Maillage généré par l'élément linéaire à 4 nœuds avec raffinement (N= 1260, E= 1176) 94
Fig. 5.22: Maillage généré par l'élément quadratique à 8 nœuds 95
Fig. 5.23: Maillage généré par l'élément quadratique à 8 nœuds avec raffinement
N= 3696, E= 1176 95
Fig. 5.24: Maillage généré par l'élément linéaire à 3 nœuds avec raffinement
N=855, E=1568 96
Fig 5. 25 : Maillage généré par l'élément linéaire à 4 nœuds avec raffinement
N= 855, E= 784 97
Fig 5. 26 : Maillage généré par l'élément quadratique à 8 nœuds avec raffinement
N= 2493, E= 784 97
Fig 5. 27 : Maillage généré par l'élément quadratique à 8 nœuds avec raffinement
Nombre d'élément=1152 Nombre de nœud= 3552 98
Fig 5. 28 : Maillage généré par l'élément linéaire à 3 nœuds avec raffinement
Nombre de nœud= 1200 Nombre d'élément=2304 98
Fig 5. 29 : Maillage généré par l'élément linéaire à 4 nœuds avec raffinement
Nombre de nœud= 1200 Nombre d'élément=1152 99
Fig 5. 30 : Maillage généré par l'élément quadratique à 8 nœuds avec raffinement
Nombre de nœud= 4719 Nombre d'élément=1536 99
Fig 5. 31 : Maillage généré par l'élément linéaire à 4 nœuds avec raffinement
Nombre de nœud= 1591 Nombre d'élément=1536 100
Fig 5. 32 : Maillage généré par l'élément linéaire à 3 nœuds avec raffinement
Nombre de nœud= 1253 Nombre d'élément=2410 100
Fig 5. 33 : Maillage généré par l'élément quadratique à 8 nœuds avec raffinement
Nombre de nœud= 1911 Nombre d'élément=600 101
Fig 5. 34 : Maillage généré par l'élément linéaire à 3 nœuds avec raffinement
Nombre de nœud= 656 Nombre d'élément=1200 101
Fig 5. 35 : Maillage généré par l'élément linéaire à 4 nœuds avec raffinement
Nombre de nœud= 736 Nombre d'élément=675 102
Fig 5. 36 : Maillage généré par l'élément quadratique à 8 nœuds avec raffinement
Nombre de nœud= 4108 Nombre d'élément=1323 102
Fig 5. 37 : Maillage généré par l'élément linéaire à 3 nœuds avec raffinement
Nombre de nœud= 1280 Nombre d'élément=2422 103
Fig 5. 38 : Maillage généré par l'élément linéaire à 4 nœuds avec raffinement
Nombre de nœud= 1392 Nombre d'élément=1323 103

Fig 5. 39 : Poutre encastrée avec charge répartie constante 107
Fig 5. 40 : la variation de déplacement y en fonction de nb de nœuds 108
Fig 5. 41: poutre supportée aux extrémités sous l'effet d'une charge constante 111
Fig 5. 42 : variation de déplacement y en fonction de nb de nœuds 112
Fig 5. 43 : Poutre encastrée soumise à une charge constante 115
Fig 5. 44 : variation de déplacement y en fonction de nb de nœuds 116
Fig 5. 45 une représentation et dimensionnement en 2 D d'une plaque rectangulaire 118
Fig 5. 46 une représentation et dimensionnement en 2 D d'une plaque rectangulaire après la symétrie. 119
Fig 5. 47 distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=25 et NE=16 121
Fig 5. 48 distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=25 et NE=32 121
Fig 5. 49 comparaison de 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0. 122
Fig 5. 50 distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=49 et NE=36 123
Fig 5. 51 distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=49 et NE=72 123
Fig 5. 52 comparaison de 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0 124
Fig 5. 53 distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=81 et NE=64 125
Fig 5. 54 distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=81 et NE=128 125
Fig 5. 55 comparaison de 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0 126
Fig 5. 56 distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=121 et NE=100 127
Fig 5. 57 distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=121 et NE=200 127
Fig 5. 58 comparaison de 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0 128
Fig 5. 59 distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=169 et NE=144 129
Fig 5. 60 distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=169 et NE=288 129

Fig 5. 61 comparaison de 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0 130
Fig 5. 62 une représentation d'un four réel 131
Fig 5. 63 une représentation et dimensionnement en 2 D d'un four après la symétrie. 131
Fig 5. 64 distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=15 et NE=8 133
Fig 5. 65 distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=15et NE=16 133
Fig 5. 66 comparaison entre 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0 134
Fig 5. 67 distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=45 et NE=32 135
Fig 5. 68 distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=45 et NE=64 135
Fig 5. 69 comparaison entre 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0 136
Fig 5. 70 distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=91 et NE=72 137
Fig 5. 71 distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=91 et NE=144 137
Fig 5. 72 comparaison entre 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0 136
Fig 5. 73 distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=153 et NE=128 138
Fig 5. 74 distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=153 et NE=256 138
Fig 5. 75 comparaison entre 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0 139
Fig 5. 76 distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=231 et NE=200 140
Fig 5. 77 distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=231 et NE=400 140
Fig 5. 78 comparaison entre 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0 141
Fig 5. 79 distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=325 et NE=288 142

Fig 5. 80 distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=325 et NE=288 143
Fig 5. 81 comparaison entre 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0 144

Liste des tableaux
Chapitre 5

Titre ***page***

Table 5. 1: résultats des déplacements d'une poutre en acier encastrée soumise à une charge répartie constante qui utilise l'élément quadrangle à 4 nœuds 106
Table 5. 2: les résultats de déplacements d'une poutre en acier encastrée soumise à une charge répartie constante qui utilise l'élément triangle à 3 nœuds 107
Table 5. 3: résultats de déplacements d'une poutre supportée aux extrémités sous l'effet d'une charge constante qui utilise l'élément quadrangle à 4 nœuds 110
Table 5. 4 résultats de déplacements d'une poutre supportée aux extrémités sous l'effet d'une charge constante qui utilise l'élément triangle à 3 nœuds 111
Table 5. 5: résultats des déplacements d'une poutre en acier encastrée soumise à une charge constante qui utilise l'élément quadrangle à 4 nœuds 114
Table 5. 6: les résultats de déplacements d'une poutre en acier encastrée soumise à une charge constante qui utilise l'élément triangle à 3 nœuds 115
Table 5. 7: représente la température T en °C Pour la distance y varie et x = 0 . 122
Table 5. 8: représente la température T en °C Pour la distance y varie et x = 0 . 124
Table 5. 9 : représente la température T en °C Pour la distance y varie et x = 0 . 126
Table 5. 10 : représente la température T en °C Pour la distance y varie et x = 0 . 128
Table 5. 11 : représente la température T en °C Pour la distance y varie et x = 0 . 130
Table 5. 12 : représente la température T en °C Pour la distance y varie et x = 0 . 134
Table 5. 13 : représente la température T en °C Pour la distance y varie et x = 0 . 136
Table 5. 14 : représente la température T en °C Pour la distance y varie et x = 0 . 138
Table 5. 15 : représente la température T en °C Pour la distance y varie et x = 0 . 140
Table 5. 16 : représente la température T en °C Pour la distance y varie et x = 0 . 142
Table 5. 17 : représente la température T en °C Pour la distance y varie et x = 0 . 144

1. Introduction générale :

De nombreux problèmes physiques se modélisent sous forme **d'équations aux dérivées partielles**. Lorsque l'on ne peut avoir directement de solutions analytiques, des méthodes numériques permettent d'obtenir une solution approchée. Ces méthodes développées depuis les années 1950, prennent en compte des modèles qui se rapprochent de plus en plus du système physique. Le domaine d'applications de ces méthodes s'étend de plus en plus grâce au progrès extraordinaire et rapide de l'informatique ; on cite par exemple la mécanique des fluides, la météorologie, la résistance des matériaux, l'électromagnétisme etc.

Dans le domaine du génie mécanique soit en mécanique des structures ou en mécanique des fluides et transfert thermique, la simulation numérique complète et même réduit les tests expérimentaux, diminuant ainsi très sensiblement les coûts de la conception des produits, d'autant plus que les performances des ordinateurs ne cessent de s'accroître et les prix ne cessent de baisser de sorte que le coût d'un calcul décroît constamment en fonction du temps.

Une étude numérique permet d'une part de réduire le nombre des essais et d'autre part, d'évaluer rapidement et quasi exhaustivement les performances de l'objet du problème traité et à moindre coût.

Dans ce contexte, on rencontre en pratique plusieurs méthodes numériques puissantes de simulation numérique pour un problème physique donné ; en occurrence : la méthode des différences finies, la méthode des éléments finis et la méthode des volumes finis. Toutes ces méthodes nécessitent l'utilisation d'un maillage discrétisant le domaine physique de calcul en points nodaux. On s'intéresse, dans le présent travail, de la méthode des éléments finis.

Cependant, la qualité des résultats obtenus dépend étroitement de la qualité du maillage utilisé. La solution est d'autant plus précise que le maillage du domaine de calcul est bien adapté au problème à résoudre. Différentes techniques peuvent être employées afin de produire des maillages de type éléments finis.

Le choix d'une technique ou d'une autre est souvent dicté par des exigences liées au degré de complexité du problème étudié et notamment par sa géométrie, mais aussi par le type du solveur utilisé ensuite pour résoudre le système d'équations du modèle.

Pour des domaines de géométries complexes, les numériciens ont de plus en plus recours à des méthodes produisant des maillages non structurés. Mais quelle que soit la méthode envisagée, il faut être conscient de la difficulté que représente le développement d'un algorithme fiable et efficace pour l'obtention de maillages répondant aux exigences de

l'utilisateur. De ce fait, la génération automatique de maillages est devenue une spécialité à part entière de l'analyse numérique.

La méthodologie générale d'une simulation numérique comprend les étapes suivantes :

- Discrétisation du problème initial,
- la solution est recherchée en certains points, appelés nœuds, du domaine de calcul,
- puis elle est interpolée sur le reste du domaine.

Pour réaliser cette opération, le domaine étudié doit être préalablement découpé en éléments simples dont les sommets sont les nœuds de la discrétisation : Ces éléments sont des polyèdres, petits volumes de l'espace ou surfaces dans le plan. C'est le maillage du domaine de calcul.

On peut distinguer deux familles de maillages :

a) Le maillage structuré :

Il est composé d'un ensemble de quadrangles ou de triangles (respectivement d'hexaèdres et de tétraèdres) empilés en couches régulières. Sa caractéristique principale est d'avoir un nombre constant de voisins par sommet. Le voisinage d'un nœud est alors connu par simple lecture matricielle, ce qui allège les structures de données.

Cependant, il n'est adapté qu'à des cas simples puisqu'il nécessite un redécoupage par bloc qui devient fastidieux dès que le domaine à étudier devient complexe.

De plus comme le nombre de voisins par sommet est constant, il n'est pas possible de procéder à des raffinements locaux, sans perdre la propriété de conformité. On ne peut donc augmenter le nombre de sommets dans les zones jugées importantes sans généraliser cet ajout à l'ensemble du domaine.

b) Le maillage non structuré :

Il est en général, constitué de simplexes (triangles en deux dimensions et tétraèdres en dimension trois) mais peut également contenir des quadrangles en dimension deux et des hexaèdres ou des prismes pour la dimension trois. Le maillage dans ce cas est dit mixte.

L'utilisation de simplexes présente de nombreux avantages. Ces polyèdres élémentaires permettent de découper des domaines très complexes. Outre cette capacité de discrétiser des formes quelconques, le maillage de type non structuré offre également une grande souplesse pour contrôler localement le nombre de points de discrétisation.

2. Problème de la génération de maillage automatique :

La littérature montre que la méthode des Éléments Finis est un outil puissant d'analyse applicable à de nombreux domaines d'études. Une des étapes essentielles à la réussite d'une analyse par éléments finis est l'étape de la génération de maillage.

Lorsque cette étape est effectuée manuellement, elle nécessite beaucoup de temps. Des erreurs et imperfections affectant la qualité des résultats peuvent facilement s'introduire, notamment à cause de la grande quantité de données à générer et à manipuler. Afin de réduire l'impact de ces problèmes, et ainsi de faciliter l'utilisation des éléments finis, de nombreuses équipes de recherche tentent de développer des méthodes visant à automatiser le processus de génération de maillage et même de l'éliminer complètement.

Dans ce contexte, le présent travail propose un code mailleur 2D en éléments finis développé en Fortran 77. L'objectif est de faciliter la phase de préparation des données en pré-traitement et répondre ainsi aux besoins urgents d'un processeur, qui est généralement commerciale très couteux et même non disponible.

Lors du développement de ce code mailleur, on a adopté une structure modulaire afin de le rendre efficace et facile à améliorer par l'utilisateur selon les exigences.

L'utilisation de plusieurs types d'élément fini et la possibilité de la division de la structure à traiter en plusieurs zones de propriétés indépendantes, lors de l'entrée des données, représentera une richesse importante pour notre code mailleur.

3. Contenu du mémoire :

Pour ce faire, on a divisé notre travail en cinq chapitres :

- Le chapitre 1 présente une synthèse bibliographique sur les méthodes de maillage effectué par des travaux antérieurs disponibles. On citera quelques applications des générateurs de maillage les plus utilisés pour la description de l'importance du maillage dans le domaine de la modélisation.
- Le chapitre 2 présente les différentes méthodes de génération de maillage en éléments finis. Après une revue bibliographique de la plupart des travaux dans le domaine de la génération de maillage, on met l'accent sur la méthode de blocs.
- Le chapitre 3 est consacré à la méthode des blocs pour la génération de maillage en 2D
- Le chapitre 4 présentera le programme et la procédure numérique avec laquelle le

code de fortran 77 traite les différents problèmes abordés.

- Enfin, les résultats seront analysés, interprétés et discutés dans le dernier chapitre 5.

Une conclusion générale résumera tous les résultats des différents cas traités et donnera les perspectives futures de développement à la lumière du présent travail.

1. Etat de l'art :

1.1. Historique de génération de maillage :

La complexité des géométries rencontrées dans les applications pratiques en mécanique ou dans des problèmes de champs en général sont telle qu'il existe un besoin bien défini pour le traitement de frontières irrégulières [1]. Ceci est particulièrement la situation dans la simulation numérique d'écoulements fluides dans les composantes de turbomachines ou les configurations de fuselage d'aile en aérodynamique externe [2].

Historiquement, on peut retracer les tentatives d'aborder de tels problèmes aux développements des techniques de transformations conformes ou plus récemment, aux méthodes de transformations algébriques [3]. Cependant, les approches généralisées pour le traitement de géométries arbitraires en trois dimensions remontent au début des années '70 avec les travaux Joe F. Thompson sur les la génération de maillages adaptés [4]. Originalement destinés aux méthodes de différences finies, ces techniques de génération de maillages s'appliquent aussi bien aux discrétisations par volumes finis ou par éléments finis [5].

Avec de tels maillages, il a été possible de satisfaire des exigences géométriques particulièrement difficiles, et des applications vraiment remarquables ont été réussies. En utilisant des techniques de décomposition de domaines, des configurations de plus en plus complexes sont possibles. Les maillages non structurés ont permis une discrétisation encore plus fidèle de la géométrie [6]. Le niveau suivant dans l'évolution de la fonctionnalité a été l'adaptation du maillage aux caractéristiques de la solution [7]. Bien que cela soit possible en partie en utilisant les maillages ajustés avec des techniques de subdivision zonale et par la concentration, une adaptation véritable par raffinement local ne peut effectivement être réalisée qu'avec les maillages non structurés [8].

En outre, les méthodes de génération de maillages ont évoluées sous l'influence des besoins des applications industrielles, mais leur développement a suivi les possibilités des techniques de modélisation géométrique, des environnements informatiques et les méthodes de simulation numériques [9].

Chronologiquement, le cheminement dans ce domaine s'est fait dans un ordre de

complexité croissante [10] :

maillages cartésiens réguliers ;
maillages cartésiens concentrés ;
maillages curvilignes ;
maillages curvilignes composites ;
maillages non-structurés ;
maillages non-structurés non-simplement connexes ;
maillages hybrides ;
maillages mobiles ;
maillages avec adaptation dans la taille et l'élancement.

1.2 Travaux réalisés dans le domaine de la génération de maillage automatique:

Les calculs par la méthode des éléments finis impliquent beaucoup de temps et l'effort de produire des données numériques comme les numéros des nœuds, les coordonnées des points nodaux et les valeurs des paramètres et la connectivité d'élément. Ainsi, les schémas de génération de maillage automatiques sont très utiles et importants pour les applications d'élément finis, particulièrement quand un grand nombre d'éléments ou la géométrie très complexe est rencontrée.

Plusieurs travaux de recherches dans le domaine de la génération de maillage automatique ont été menés pour des éléments rectangulaires et triangulaires.

En 1988, Une tentative a été faite par K. Ho-Le [11] pour fournir une image claire et complète du champ de génération de maillage par éléments finis. Il a proposé un schéma qui classifie les méthodes de génération de maillage, par lequel il a identifié sept approches importantes de génération de maillage (Fig.1). La base de cette classification est l'ordre temporel dans lequel les nœuds et les éléments sont créés. Seulement quatre approches s'avèrent être des méthodes complètement automatiques de génération de maillage. Ces approches sont comparées sur la base de critères des types et des formes d'élément, le contrôle de densité de maillage, et l'efficacité de temps.

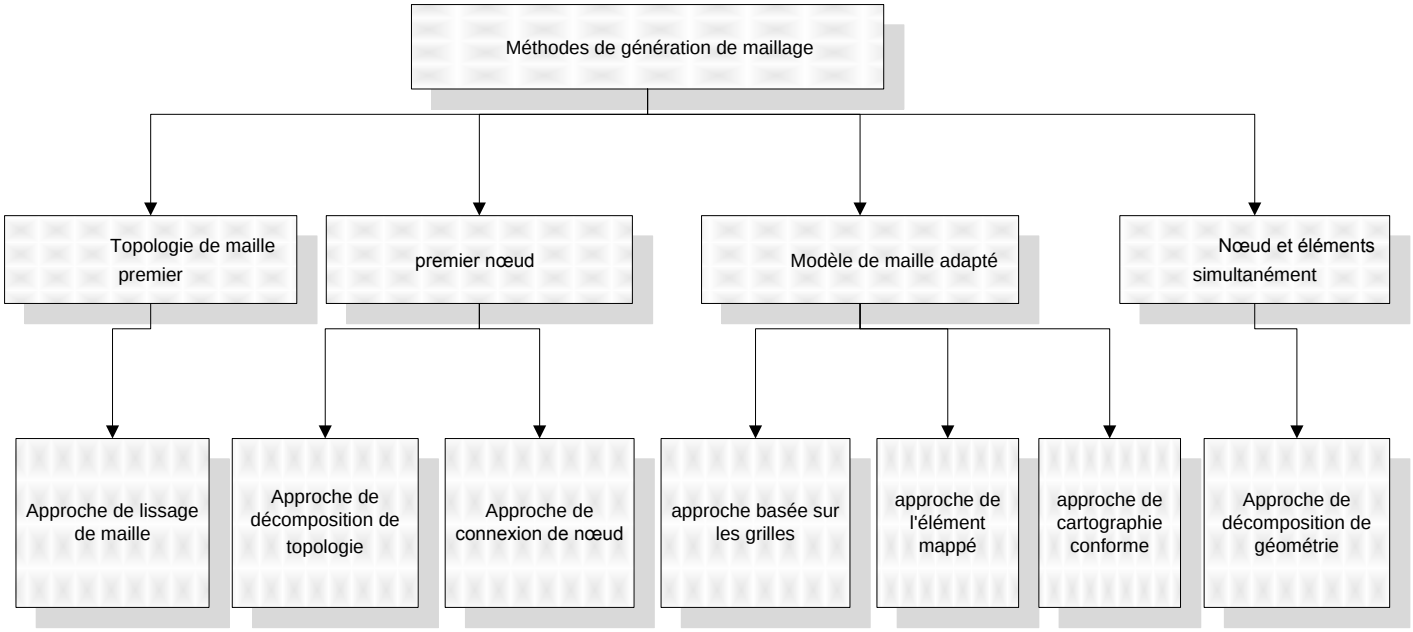


Figure 1.1 Plan de classification pour méthodes de génération de maillage [11]

En 1988, Taniguchi Takeo [12] a proposé un nouveau générateur automatique interactif de maillage pour l'utilisation sur des micro-ordinateurs. La méthode proposée (appartenant à l'origine à la méthode de blocage) génère des propriétés topologiques indépendamment d'autres propriétés géométriques et physiques qui sont exigées pour l'analyse par les éléments finis. Elle inclut un générateur efficace de maillage pour des régions transitoires avec des configurations quadrilatérales et triangulaires, et elle peut augmenter considérablement la flexibilité de l'application de la méthode à une région avec une configuration complexe et, en même temps, elle peut diminuer la somme des données exigées à l'entrée pour la génération de maillage. Actuellement, cette méthode est valable seulement pour des cas bidimensionnel.

En 1991, Les techniques de génération des mailles orthogonales bidimensionnelles sont discutées par Kai-Lung Lin et Heiu-Jou Shaw [13] qui ont présenté deux méthodes différentes pour la mise en œuvre de la transformation de coordonnées. La première méthode génère des mailles permettant le déplacement libre de nœuds le long de la frontière, et l'autre construit des mailles avec les nœuds prescrits sur la frontière. Pour la première méthode, le module conforme ou facteurs d'échelle ne doivent pas être calculé comme les méthodes précédentes. Pour la deuxième méthode, les fonctions de contrôle sont itérativement ajustées en fonction de l'angle de l'intersection des lignes de la maille. Dans ce processus, les fonctions de commande sont considérées comme partie de solutions et ne peuvent pas

continuer à contrôler l'espacement entre les lignes de maille.

Des calculs numériques sont réalisés pour illustrer la détermination de différents types de distribution de points de maille sur la frontière. Les méthodes pour des parties de points de maille fixes sur la frontière sont également fournies pour des applications plus larges.

En 1992, une procédure simple et précise a été présentée par V.N. Kaliakin [14] pour déterminer les coordonnées des points de nœud situés à l'intérieur des régions planaires discrétisées avec des éléments quadrilatéraux. Cette procédure a surmonté beaucoup de défauts associés aux variantes de la méthode de génération de super élément.

Dans la même année, P.T.S. Ho [15] a présenté un programme informatique de génération automatique de maillage par élément fini triangulaire à trois nœuds ou à six nœuds basée sur des fonctions de cartographie isoparamétrique. Le programme est écrit dans le FORTRAN 77.

Étant donné les frontières de la surface de la structure, ce programme discrétise l'espace interne dans grilles régulières de densité définie pour l'utilisateur et de coordonnées nodales de produit et la connectivité nodale avec la réduction facultative de profil de largeur de bande pour une analyse efficace par éléments finis.

En 1993, Yuan-Chong Zhang, Jia-Ju Yin et Lei Luo [16] ont présenté un générateur de maillage bidimensionnel avec des conditions aux limites générées automatiquement qui est une version améliorée d'un programme d'ordre variable de génération de maillage par Liu et Chen. Les améliorations comprennent l'aptitude à générer une série de points ou des contraintes de face et des charges sur la limite des éléments de nœuds générés ou des éléments de côtés à partir d'informations sur le super élément original. Un programme Fortran est donné afin de générer des mailles quadrilatéraux et conditions aux limites différentes pour un problème axisymétrique.

En 1994, un nouvel arrangement pour la génération d'une maille quadrilatérale d'élément a été présenté par C.K. Lee et S.H. Lo [17]. L'algorithme se sert du fait qu'une maille triangulaire d'élément liée par un chiffre pair de lignes segments peut toujours être convertie en quadrilatères.

En employant la technique d'avancement en avant pour fusionner schématiquement

des éléments triangulaires, des mailles quadrilatérales bien-évaluées de haute qualité peuvent être formées sans n'importe quel traitement pénible pour les triangles d'isolement.

À la différence de beaucoup d'autres méthodes, aucune coupe-ligne ou division manuelle du domaine traité dans des sous-régions plus simples n'est exigée avant conversion. Le nombre et la position des nœuds de frontière ne sont pas changés pendant le procédé de génération de maille. Différentes régions matérielles partageant les lignes de frontière communes peuvent être traitées individuellement. Ce processus peut être appliqué à n'importe quel domaine 2D arbitraire avec ou sans ouvertures internes.

En 1995, H. Borouchaki, George, P. L, F. Hecht, P. Laug et E. Saltel [18] ont présenté quelques résultats obtenus par une méthode de maillage de type Delaunay dans le cas où cette dernière est gouvernée par une carte métriques (isotropes ou anisotropes) donnée. Leur travail constitue la deuxième partie de leur étude, la première partie décrivant les aspects algorithmiques de la méthode de maillage. Des exemples académiques ont été donnés puis ils ont montré quelques applications en mécanique des fluides.

En 1996, M. H. Kadivar et H. Sharifi [19] ont étudié les méthodes de génération de maillage isoparamétrique curviligne conventionnelle qui produisent normalement des mailles bien conditionnées. C'est dû principalement du fait que le nombre de nœuds des bords opposés de super-élément devrait être égal. La méthode isoparamétrique est améliorée pour des problèmes bidimensionnels et tridimensionnels, tels que les bords opposés peuvent avoir un nombre différent des nœuds.

La méthode isoparamétrique prolongée donne beaucoup plus de flexibilité en choisissant le nombre de nœuds et semble prometteuse pour être appliquée aux différentes méthodes de génération de maillage adaptative. Une méthode semblable peut également être appliquée dans un générateur de maillage multifonctions.

En 1997, Borouchaki Houman, Frey Pascal J, George Paul Louis [20] ont traité, en premier lieu, une méthode de génération de maillage adaptative gouvernée par une carte discrète de métriques (isotropes ou anisotropes), basée sur la génération de maillages unité par rapport à une structure riemannienne. Ils ont proposé ensuite un schéma de conversion automatique des triangles en quadrangles, généralisant le cas classique dans un contexte anisotrope. Quelques exemples d'applications, en particulier en mécanique des fluides ont

illustrés l'efficacité de la méthode.

Dans le présent travail on a reprise les routines des travaux [3,14,15,17] pour développer un code de structure modulaire .des modifications sont introduits afin de donner au code développé une structure cohérente .une vérifications des résultats obtenus par [3,15] seront vérifiés.

Par ailleurs, la méthode de maillage par blocs, est présentée pour produire des éléments d'ordre variables par la génération de super-éléments. Cette méthode a l'avantage d'être simple à appliquer et qu'elle exige moins de temps d'exécution en comparaison avec les autres générateurs de maillage.

Selon la complexité géométrique et la nature du matériau étudié, les super éléments sont manuellement déterminé afin d'être raffiné dans des éléments d'ordres divers.

Plusieurs sous-programmes de génération de maillage ont été employés pour produire des éléments d'ordres différents et fournir ainsi les coordonnées des points nodales, les valeurs de paramètre et la connectivité de chaque élément.

Le code mailleur développé lors de cette étude est adapté à un solveur de la bibliographie (MAT-fem) [21] pour des applications dans le domaine solide et thermique d'une part et au code de Reddy [22] d'autre part afin de mettre en évidence l'efficacité et les performances de ce mailleur développé.

2. MÉTHODES DE GÉNÉRATION DE MAILLAGE :

Il n'est pas question de donner ici une liste exhaustive de toutes les méthodes de maillage existantes. Elles peuvent être cependant divisées en deux types :

D'une part, les méthodes créant des *maillages structurés* encore appelés *grilles* ; d'autre part, les méthodes générant des *maillages non structures*.

Signalons que pour certaines applications, des méthodes des deux types sont utilisées conjointement afin d'obtenir des maillages hybrides comportant des régions où le maillage est structuré et des régions où ce maillage est non structuré [23].

La différence entre *maillages structurés* et *maillages non structurés* réside dans leur connectivite, c'est à dire dans le type de connexions autorisées entre leurs sommets. Voici une définition pour caractériser les maillages structurés ou non structurés [24] :

Un maillage est dit structuré si sa connectivité est de type différences finies ; il est dit non structuré si sa connectivité est de type quelconque [24].

2.1 Les maillages structurés :

Un maillage structuré se distingue par un arrangement des éléments de maillage en un ou plusieurs blocs (ou groupes) selon un certain nombre de directions logiques. Par exemple, en trois dimensions, un maillage complètement structuré est composé de blocs d'éléments hexaédriques disposés cote à cote selon trois directions logiques [25]. Les frontières géométriques sont définies par les limites externes des blocs et peuvent être ainsi transformées en une grille de calcul cubique dans le solveur [26].

Les principaux avantages des maillages structurés tiennent à l'efficacité de stockage des données et à la précision accrue fournie par les éléments très étirés. Les données conservées sur un maillage structuré peuvent en effet être disposées en mémoire de façon à profiter de la structure implicite fournie par le maillage, et ne nécessitent donc pas l'utilisation d'un système explicite d'indexage comme pour les maillages non structurés [27].

Par ailleurs, un maillage structuré fournit une précision accrue, pour un nombre de sommets et d'éléments donné, en permettant l'utilisation des éléments ayant un très

grand élancement [28]. Des éléments très étirés permettent de bien capturer certains phénomènes anisotropes, comme, par exemple, les couches limites près des parois solides. II existe plusieurs algorithmes de génération de maillages structurés, qui sont maintenant bien connus et documentés (Thompson et al. (1985)) [4]. Tous ces algorithmes sont caractérisés par un modèle de transformation d'un espace physique en un espace logique à l'aide d'une relation soit connue, soit issue d'une équation aux dérivées partielles.

Les principales méthodes de génération de maillages structurés sont [29]:

- la méthode algébrique,
- Méthodes par résolution d'EDP
- Méthodes de blocs
- la méthode elliptique,
- la méthode hyperbolique.

En général, il est impossible de générer un bon maillage autour ou à l'intérieur d'une structure industrielle typique sans utiliser quelques dizaines, voire quelques centaines de blocs [30]. Le principal problème inhérent à l'utilisation de maillages structurés tient donc à la tâche colossale que représente la définition des blocs nécessaires à l'initialisation du processus de maillage [31].

Certains outils interactifs relativement généraux font leur apparition sur le marché pour la génération de tels maillages . De nombreux logiciels sont disponibles sur le marché utilisant ce type de générateurs de maillage (Abaqus, I-deas ,Adina ,Marc, Algor , Radioss ,Ansys Samcef , Catia , Systus) [32].

Il reste que leur utilisation demande du temps et de l'expérience, et que des modifications au modèle géométrique, même relativement simples, entraînent souvent un travail considérable de reconnexion et de redéfinition des blocs **[33].**

2.1.1 Méthodes algébriques :

2.1.1.1 Introduction :

Les **méthodes algébriques**, encore appelées méthodes de **transport-projection**, se conçoivent comme des applications d'un maillage dans un **espace paramétrique (u^1, u^2, u^3)**

appliquées vers **l'espace physique (x_1, x_2, x_3)**. Cette transformation est réalisée par une **expression algébrique explicite.**

Cette approche est particulièrement facile et efficace puisque le maillage est généré comme un maillage cartésien régulier dans l'espace paramétrique. On note qu'il conserve ses propriétés topologiques, c'est-à-dire il sera **structuré.**

Le maillage dans **l'espace physique** est obtenu par l'évaluation directe des expressions de la transformation de la forme ;

$$x_i = x_i(u^1, u^2, u^3)$$

où $u = (u^1, u^2, u^3)$ représente les coordonnées dans l'espace paramétrique, et $x=(x_1,x_2,x_3)$ sont les coordonnées dans l'espace physique, tel qu'illustré à la Figure (2.1).

Le problème se ramène à trouver de telles relations qui vérifient les conditions énumérées, c'est-à-dire que la transformation réalise effectivement une application d'un rectangle topologique vers la frontière du domaine tel qu'illustré à la Figure (2.2).

Les diverses techniques de maillages algébriques sont caractérisées par le fait que l'expression de la transformation est connue explicitement ; ce qui les rend particulièrement efficaces car la génération du maillage se ramène à une simple évaluation de fonctions [34].

Parmi ces nombreuses méthodes, on retrouve les suivantes :

- Des expressions analytiques.
- Les transformations conformes.

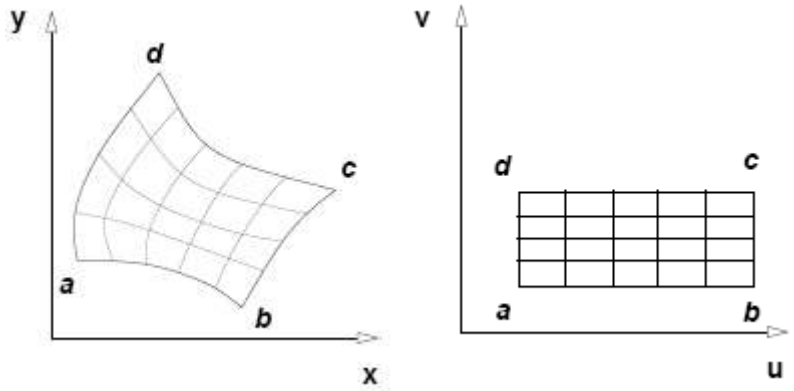


Figure 2.1. Transformation d'un maillage de l'espace paramétrique vers l'espace physique

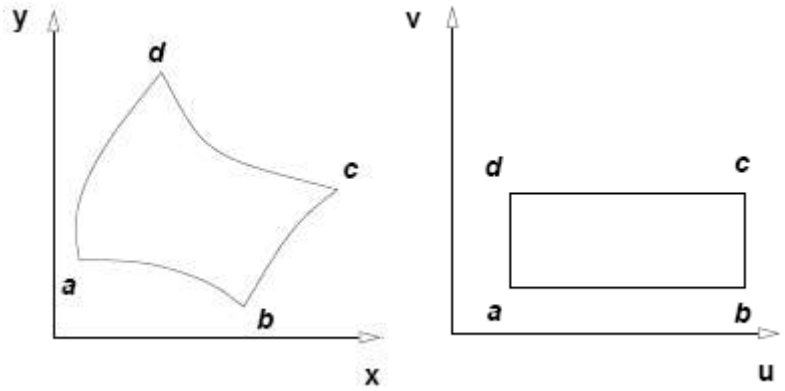


Figure 2.2. Transformation des frontières.

Des exemples dans l'utilisation d'expressions analytiques pour la génération d'un maillage sont les systèmes de coordonnées classiques. On rappelle que ce type de transformations s'applique à l'espace au complet plutôt qu'à des régions bornées par des frontières quelconques. Pour de tels domaines, ces approches ne donnent pas des maillages adaptés, et ne peuvent pas être des méthodes générales de maillages.

Semblables en principe, les transformations conformes donnent des maillages spécialisés avec une plus grande variété de possibilités, mais avec des restrictions liées au choix des fonctions de variables complexes [35].

Un exemple de l'utilisation d'une **expression analytique** pour générer un maillage est la transformation suivante :

$$x = r\cos(\theta) \qquad (2.1)$$

$$y = r\sin(\theta) \qquad (2.2)$$

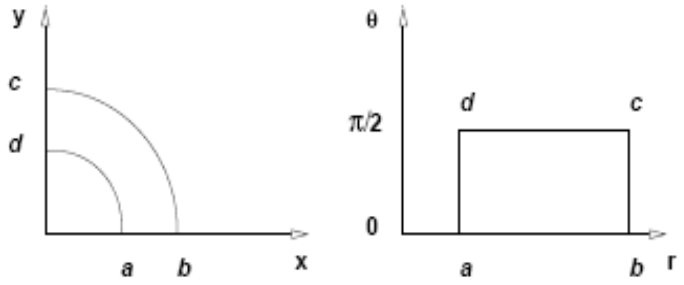


Figure 2.3. Exemple d'une transformation algébrique

Le maillage (Figure 2.3) ainsi engendré est déterminé par le choix des fonctions f=rcos(θ) et g=rsin(θ) et par conséquent cette technique ne pourra pas être utilisée pour des géométries quelconques.

En résumé, ces deux méthodes ne permettent pas un contrôle sur la géométrie des frontières [29].

2.1.1.2 Transformations conformes :

Parmi les premières techniques de transformations algébriques sont les transformations conformes qui appliquent un domaine dans **le plan complexe w vers** un domaine dans **le plan complexe z** en utilisant une **fonction d'une variable complexe** de la forme,

$$z = f(\omega) \qquad (2.3)$$

Où $z = x + iy,\ \omega = u + iv, - et.i = \sqrt{-1}$

Ces transformations sont **basées** sur **les propriétés de la fonction f d'une variable complexe qui se décompose en une partie réelle** :

$$x = x(u,v) \qquad (2.4)$$

et une partie imaginaire :

$$y = y(u,v) \qquad (2.5)$$

Les fonctions **x** et **y** ne peuvent être **choisies de façon arbitraire**, mais **doivent vérifier** les relations de **Cauchy-Riemann :**

$$\frac{\partial x}{\partial u} = \frac{\partial y}{\partial v} \qquad (2.6)$$

$$\frac{\partial x}{\partial v} = -\frac{\partial y}{\partial u} \qquad (2.7)$$

On dérive ces relations par rapport à u et v, respectivement :

$$\frac{\partial^2 x}{\partial u^2} = \frac{\partial^2 y}{\partial v \partial u}$$

$$\frac{\partial^2 x}{\partial v^2} = -\frac{\partial^2 y}{\partial u \partial u}$$

En additionnant, on obtient

$$\frac{\partial^2 x}{\partial u^2} + \frac{\partial^2 x}{\partial v^2} = 0$$

On répète l'opération en dérivant par rapport à v et u, respectivement. On obtient :

$$\frac{\partial^2 y}{\partial u^2} + \frac{\partial^2 y}{\partial v^2} = 0$$

Ce qui donne :

$$\nabla^2 x = 0 \qquad (2.8)$$

$$\nabla^2 y = 0 \qquad (2.9)$$

C'est-à-dire **x** et **y** sont des **fonctions harmoniques**. Il en découle un certain nombre de propriétés utiles dans le cadre de la génération de maillages [36].

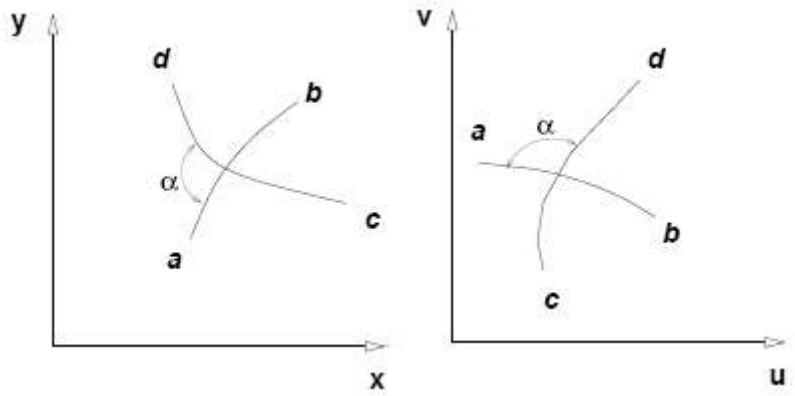


Figure2. 4 Préservation des angles par les **fonctions harmoniques.**

Ces **fonctions harmoniques**, dans le voisinage d'un point, transforment les courbes de façon semblable. En particulier, **la transformation préserve les angles entre les courbes**.

L'approche consiste à générer un maillage régulier cartésien dans le plan **ω**, et le maillage est obtenu par l'application **f** vers le plan **z**. **Le maillage résultant dépendra du choix de la fonction f**.

2.1.1.2.1 La transformation $\omega = z^2$:

À titre d'exemple on analyse la transformation $\omega = z^2$ qui applique un **point (x, y) du plan z** vers **les images (u, v) du plan ω**. La transformation peut s'expliciter en la décomposant en ses parties réelle et imaginaire $z = x + iy$, $\omega = u + iv$, *et* $i = \sqrt{-1}$.

$$\begin{aligned} u &= x^2 - y^2 \\ v &= 2xy \end{aligned} \qquad 2.10$$

La droite $y=y_0$ dans l'espace (x,y) donne la courbe paramétrique :

$$\begin{aligned} u &= x^2 - y_0{}^2 \\ v &= 2xy_0 \end{aligned}$$

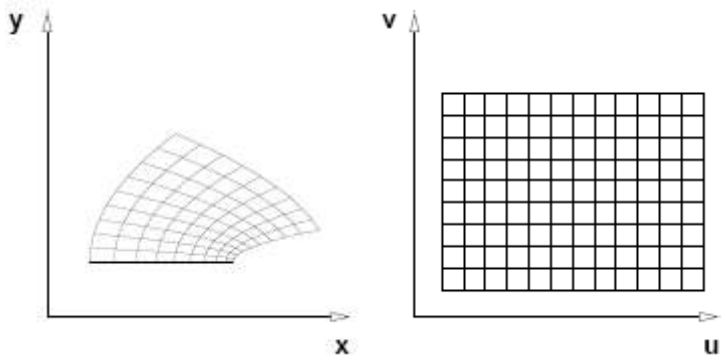


Figure 2.5. La transformation $\omega = z^2$.

Dans l'espace (u,v), en éliminant x, on obtient :

$$u = \frac{v^2}{4y_0^2} - y_0^2 \qquad 2.11$$

qui est **l'équation explicite d'une parabole**.

En variant le paramètre $y = y_o$, on obtient une famille de courbes paraboliques.

De façon semblable, la droite x = x_o dans l'espace (x, y) donne la courbe :

$$u = x_0^2 - \frac{v^2}{4x_0^2} \qquad 2.12$$

Qui est également une parabole, mais centrée sur v.

En variant les paramètres x =x_o et y=y_o, on obtient deux familles de courbes paraboliques. Comme illustré à la Figure2.5, ces deux familles sont orthogonales découlant de la propriété de la préservation des angles [10].

2.1.1.2.2 La transformation ω = e^z + z :

Une autre transformation remarquable d'intérêt dans **le domaine de l'aérodynamique** est la transformation suivante :

$$\omega = e^z + z$$

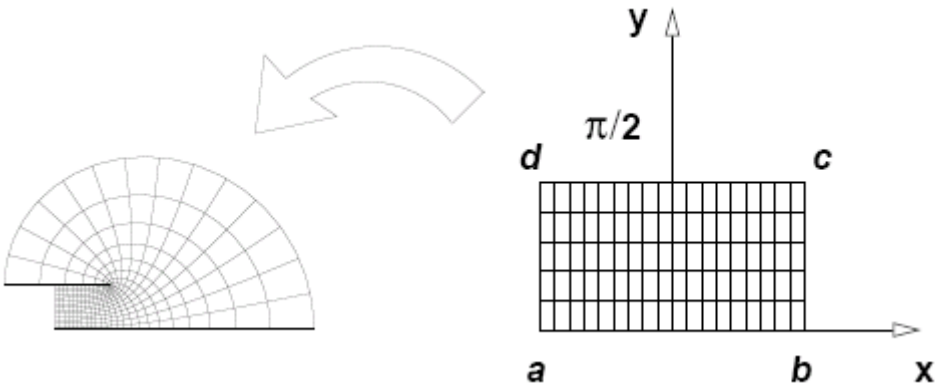


Figure 2.6. La transformation ω = e^z + z

Qui applique les parties positive et négative de la droite $y = \pi$ vers une demi-droite :

$$v = \pi, \qquad u \le -1$$

Ce type de configuration se prête à des applications d'entrée (sortie) de canalisation telle qu'il est illustré à la figure 2.6 en utilisant **la relation d'Euler** :

$$e^{i\theta} = \cos\theta + i\sin\theta$$

Et en substituant dans $\omega = e^z + z$, on obtient :

$$\omega = u + iv = e^x \cos y + ie^x \sin y + x + iy$$

Qui se décompose en ses parties réelle et imaginaire :

$$u = e^x \cos y + x \qquad 2.13$$

$$v = e^x \sin y + y \qquad 2.14$$

Pour transformer la droite $y=y_0$ du plan (x,y) vers le plan (u,v), on remplace y par y_0 ce qui donne :

$$u = e^x \cos y_0 + x$$

$$v = e^x \sin y_0 + y_0$$

On élimine la variable x en l'isolant dans l'équation de v :

$$x = \ln \frac{v - y_0}{\sin y_0}$$

Et on substitue dans l'équation de u ce qui donne :

$$u = \frac{v - y_0}{\sin y_0} \cos y_0 + \ln \frac{v - y_0}{\sin y_0} \qquad 2.15$$

On remarque que cette expression est indéterminée si le dénominateur $\sin(y_0)$ est nul; ce qui correspond à $y_0 = m\pi$ pour des valeurs de m=0,1,2,….. on traite les cas où m est pair et impair séparément :

m est pair on a alors :

$$u = e^x + x$$

$$v = y_0$$

C'est une droite $v = y_0$ avec u variant de moins l'infini à pus infini. On prendra m=0, donnant $y_0 = 0$ qui se transforme en la droite v=0.

m est impair cela donne :

$$u = -e^x + x$$

$$v = y_0$$

On a toujours une droite, mais la variation de u atteint un extremum à x = 0, donc à u = -1. En fait, il s'agit d'une demi-droite allant de moins l'infini à u = -1. On peut interpréter ce cas comme la droite $y = y_0$ du plan (x, y) étant pliée à u = -1. On prendra m =1, donnant $y_0 = \pi$ qui se transforme en la demi droite $v = \pi$ de moins l'infini à u=-1.

La deuxième famille de courbes peut être trouvée en posant la droite $x = x_0$ dans le plan (x, y). Ce qui donne [37]:

$$u = e^{x_0} \cos y + x_0$$

$$v = e^{x_0} \sin y + y$$

On isole **y** dans la première équation :

$$y = \cos^{-1}\left(\frac{u - x_0}{e^{x_0}}\right) \qquad 2.16$$

Et on remplace dans la deuxième :

$$v = e^{x_0} \sin\left(\cos^{-1}\left(\frac{u - x_0}{e^{x_0}}\right)\right) + \cos^{-1}\left(\frac{u - x_0}{e^{x_0}}\right) \qquad 2.17$$

Cette équation est définie seulement pour les valeurs d'argument $\frac{u - x_0}{e^{x_0}}$ valides dans le cosinus inverse, de -1 à 1, donc sur la plage $-e^{x_0} + x_0$ à $e^{x_0} + x_0$.

Cette analyse donne les plages des paramètres u et v :

$$1.5\pi < u < 1.75\pi$$
$$0 < v < \pi$$

Une caractéristique des **transformations conformes** est la préservation des angles. Donc, le maillage dans le plan **paramétrique** étant cartésien, le maillage transformé sera orthogonal.

De par leurs origines mathématiques, les transformations conformes se prêtent mal à des maillages concentrés.

Finalement, comme transformations d'une variable complexe (à deux composantes) le passage à la dimension trois n'est pas possible, ce qui limite leurs applications à des géométries 3d.

La Figure 2.7 montre deux exemples obtenus pour des domaines respectivement analogues à un quadrangle et à un triangle. Malgré la non convexité de ces domaines, les maillages sont valides et de qualité acceptable.

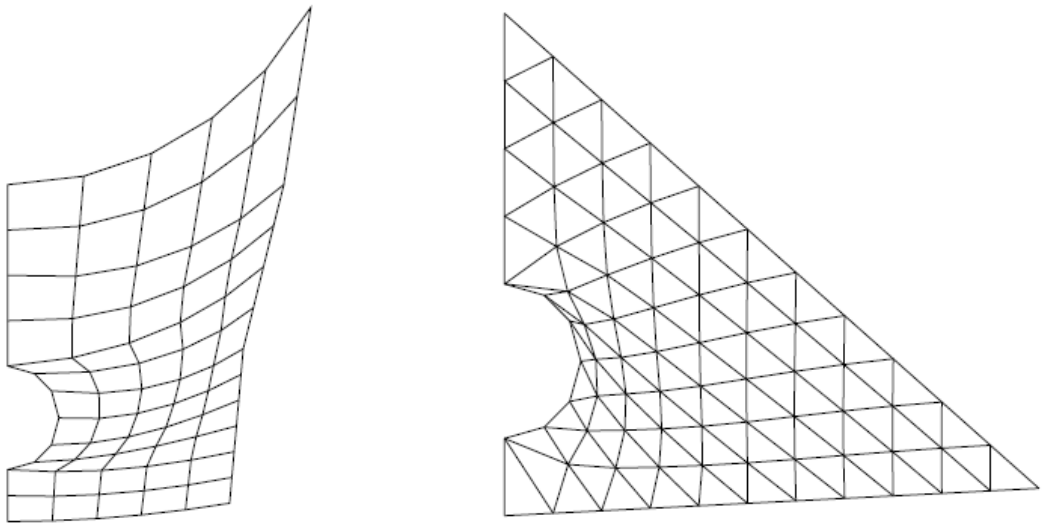


Figure 2.7 Maillages algébriques, sur un domaine «quadrangulaire » (à gauche) et « triangulaire » (à droite).

Les méthodes algébriques sont donc naturelles et pratiques dès lors que le domaine considéré présenté une **forme assez simple**. Pour des objets de géométrie complexe, on peut recourir à ce type de méthode à condition d'avoir découpe au préalable le domaine à mailler en sous domaines dont la géométrie est plus élémentaire. C'est l'objet des méthodes dites **« de**

blocs » [34].

2.1.2 Méthodes de blocs :

Les méthodes de blocs, encore dénommées méthodes de super-éléments, sont la **généralisation des méthodes algébriques** à des **géométries plus complexes**, Le domaine à mailler est partitionné en blocs dont la géométrie doit être suffisamment simple afin qu'ils puissent être mailles par les techniques précédentes. Ces méthodes sont semi-automatiques dans la mesure où elles nécessitent une définition le plus souvent manuelle de la partition en blocs du domaine [38].

D'autre part, **il faut veiller à ce que le maillage global, obtenu après assemblage des maillages des différents blocs, soit toujours correct, essentiellement aux frontières entre blocs.** Notons par ailleurs que ces méthodes autorisent l'utilisation de différents types d'éléments pour mailler les blocs, à condition que la connectivité reste correcte. On obtient alors un maillage mixte [39].

Ce type de méthode trouve une utilisation pour des objets dont le partitionnement est naturel. On applique la méthode de bloc en 2D pour chaque bloc du maillage grossier, sur l'approche. Le maillage résultatant **(figure 2.8)** comprend 366 éléments et 438 nœuds. Le maillage grossier (i.e. l'ensemble des blocs de départ) consiste en 11 éléments, de type quadrangle, construits à partir de seulement 18 points [34].

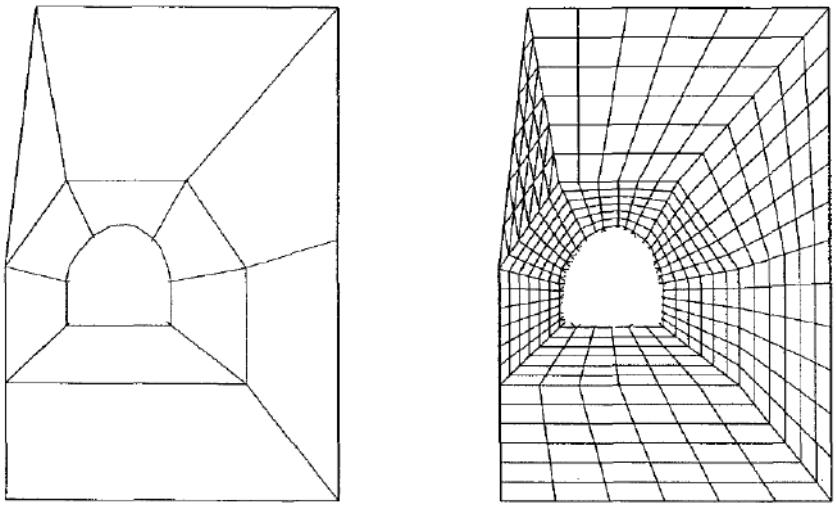


Figure **2.8**: Maillage grossier et maillage final

2.1.3 Méthodes par résolution d'EDP :

À l'instar des **méthodes algébriques**, ces méthodes s'appliquent à des **domaines de géométrie simple**, analogues à des quadrangles en dimension deux ou des hexaèdres en dimension trois. La différence se situe dans le choix de la fonction servant à générer le maillage. Au lieu d'être connue par avance, **celle-ci est déduite de la résolution d'un système d'équations aux dérivées partielles judicieusement choisi.** Suivant le type d'opérateurs utilisés dans ce système, la méthode est dite **elliptique**, **parabolique** ou **hyperbolique**. Ces méthodes permettent éventuellement d'imposer **certaines contraintes** sur les **maillages à générer**, par exemple l'orthogonalité entre les mailles. On trouvera une étude détaillée de ces méthodes et de leurs applications par exemple dans Thompson et al. 1985 [4].

La Figure 2.9 est un exemple de maillage obtenu par une méthode elliptique.

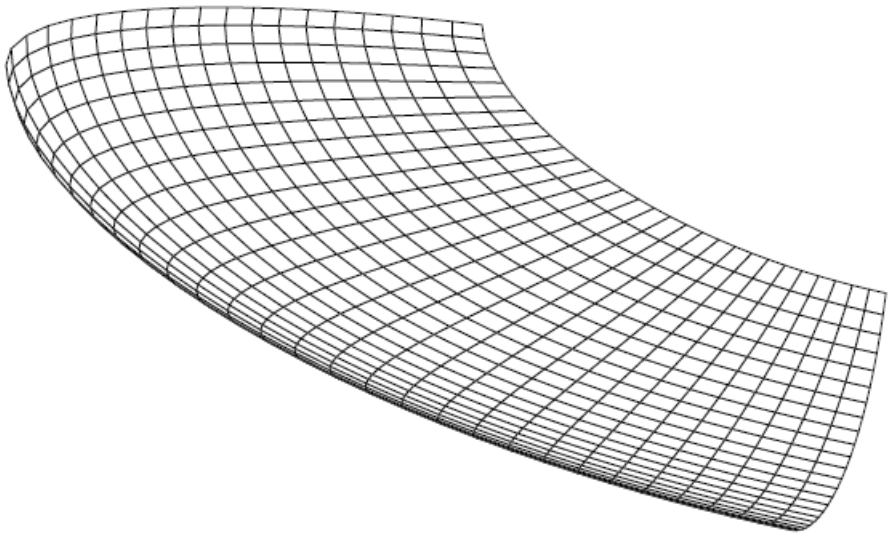


Figure 2.9 Exemple de maillage par une méthode elliptique.

2.1.4- Autres méthodes :

Parmi les autres méthodes existantes, on peut mentionner celles propres à des topologies particulières. Ainsi, à partir d'un maillage bidimensionnel de référence, il est possible d'obtenir un maillage tridimensionnel par empilement d'éléments suivant un axe pour une topologie hexaédrique ou autour d'une génératrice pour une topologie cylindrique (méthode d'extrusion) [34].

2.2 Les maillages non structurés :

La seconde approche de génération de maillage est l'approche non structurée. Celle-ci vise à remplir l'espace à mailler en plaçant les éléments sans tenter de maintenir de relation d'ordre entre les voisins; un maillage non structuré ne peut donc pas être transformé en une grille de calcul régulière (structurée) au niveau du solveur.

Dans la majorité des cas, les algorithmes de génération de maillages non structurés construisent des éléments basés sur des formes géométriques de type simplex (par exemple des tétraèdres en trois dimensions), mais il existe également quelques algorithmes de génération de maillages non structurés utilisant des quadrilatères (R. Löhner and P. Parikh ,(1988) [2] ou O. Hassan, N.P. Weatherill, J. Peraire, K. Morgan, E.J. Probert (1995) [8]).

Les techniques de génération de maillages non structurés se regroupent en trois principales classes d'algorithmes :

- la méthode de Delaunay (H. Borouchaki, P.L. George and S.H. Lo (1996) [40])
- et la méthode de l'avance frontale (J. Frykestig (1994) [41])
- et Méthodes de quadtree-octree (M.A. Yerry and M.S. Shephard (1984) [42])**.**

La première, qui repose sur un fondement mathématique très rigoureux, semble maintenant l'emporter en fiabilité et en performance sur la seconde. Ces deux méthodes sont certainement de loin les plus populaires, mais la recherche demeure très active dans ce domaine, et les méthodes de génération de maillages non structurés combinant différents aspects des méthodes de Delaunay et de l'avance frontale sont aussi utilisées (J.D. Muller, P.L. Roe and H. Deconinck (1992) [43]).

Contrairement aux méthodes de génération de maillages structurés, l'approche non structurée permet d'automatiser facilement le traitement de géométries même très complexes, et les maillages ainsi générés se prêtent bien au remaillage local ou à l'adaptation de maillage [44]. Par contre, la résolution des équations de Navier-Stokes reste difficile sur ce type de maillage, en particulier à cause des élancements extrêmement élevés nécessaires dans les éléments près des parois, et de façon générale dans les régions où l'écoulement est fortement cisaillé. La précision des résultats dépend en effet, dans une large

mesure, du contrôle des angles maximum à l'intérieur des éléments (O. Hassan, N.P. Weatherill, J. Peraire, K. Morgan, E.J. Probert (1995) [8]) et du rapport de taille entre les éléments adjacents.

Cependant, ces deux critères de qualité du maillage sont plus difficiles à contrôler dans le cas des maillages non structurés. En effet, la convergence d'une méthode de maillage est sa capacité à générer des maillages valides.

2.2.1 Méthode de type Delaunay-Voronoï :

Depuis le début des années 80, les propriétés des polyèdres de Voronoï sont utilisées pour générer des maillages et sont toujours le sujet de nombreuses publications (M. Berger (1978) [45] et D.J. Mavriplis (1992) [46] et H. Borouchaki, P.L. George et S.H. Lo (1996) [40]). Leurs principes reposent sur les propriétés du diagramme de Voronoï qui permet de construire un maillage en simplexes de l'enveloppe convexe d'un nuage de points. Cependant cette construction suppose que les points qui seront les sommets du maillage sont connus. Généralement, la solution obtenue contient trop peu de sommets pour qu'un calcul numérique puisse être effectué. Il Faut alors enrichir la discrétisation du domaine de calcul. De plus, la méthode construit le maillage de l'enveloppe convexe d'un nuage de points ; la frontière n'est donc pas forcément respectée pour un domaine quelconque[47].

Trois étapes sont requises pour générer le maillage d'un domaine quelconque :

1. Création de la triangulation Delaunay d'un nuage de point.

2. Création de points intérieurs pour obtenir un maillage de Delaunay utilisable par le calcul numérique.

3. Modification éventuelle du maillage pour respecter la frontière du domaine.

Avant de décrire le mailleur de Delaunay, nous donnons quelques résultats préliminaires démontrés dans (George et Borouchaki (1997) [29]).

2.2.1.1 Triangulation de Delaunay :

Regardons d'abord l'existence et la construction d'un maillage à partir d'un nuage de points. Soit donc ***N*** **un ensemble fini de points** $(S_i)1\leq i\leq n$ de $\mathbf{R}^d$ [48] .

Avant de parler de triangulation de Delaunay, il faut introduire le diagramme de Voronoï :

– **un d-polytope convexe** est **l'enveloppe convexe de plusieurs points** de $\mathbf{R}^d$ (c'est la généralisation d'un **polygone convexe** en 2D) ;

– **le diagramme de Voronoï de *N*** est **l'ensemble des d-polytopes convexes** (V_i) $1 \leq i \leq n$ appelés **cellules de Voronoï** et définis par :

$$V_i = \{P \in R^2 \mid .\ d(P, S_i) \leq d(P, S_j),\ \forall j \neq i\ \} \qquad (2.18)$$

C'est-à-dire l'ensemble des points les plus proches du point S_i que des autres points S_j [49].

Le diagramme de Voronoï présente des propriétés fondamentales particulièrement intéressantes.

Proposition :

– les cellules de Voronoï recouvrent R^d sans chevauchement (Figure 2.10 à gauche)

– si les points de *N* sont d+2 à d+2 **non cocycliques** (**c'est-à-dire non alignés sur la même boule**), alors **le dual *T* de diagramme de Voronoï** ne contient que des **simplexes** ;

– sinon le dual contient des **d-polytopes** que l'on peut décomposer en simplexes et obtenir à nouveau une triangulation *T* ;

– ***T*** est appelé **triangulation de Delaunay** et forme avec *N* **un maillage de l'enveloppe convexe de *N*** (Figure 2.10 à droite) [50].

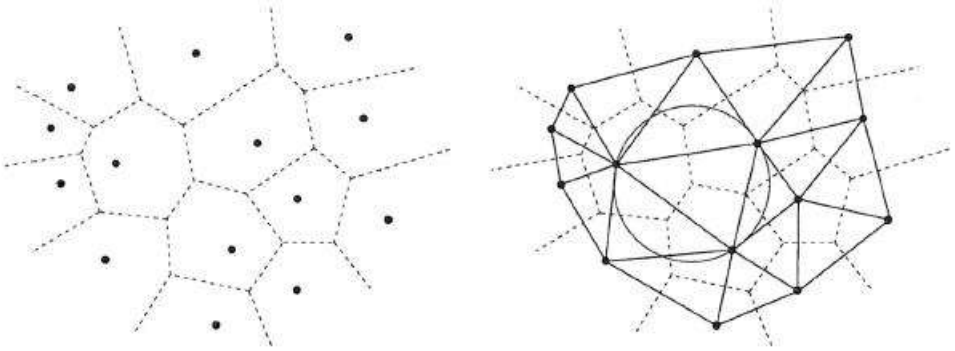


Figure **2.10**– **Diagramme de Voronoï** d'un nuage de nœuds et **triangulation de Delaunay** correspondante en 2D.

Pour construire une triangulation de Delaunay, le diagramme de Voronoï n'est pas explicitement construit. En pratique, pour être sur d'obtenir une triangulation de Delaunay, c'est le critère suivant qui est **exactement** vérifié.

Théorème (critère de Delaunay) : soit ***T* une triangulation de l'enveloppe convexe** de ***N, T*** est **une triangulation de Delaunay** de ***N*** si et seulement **si** la **boule ouverte circonscrite** de chaque élément ne contient aucun nœud.

Le problème est que **Ω** correspond rarement à l'enveloppe convexe de ***N*** [51]. Il faut donc d'abord construire un maillage de l'enveloppe convexe de Ω a l'aide du théorème précédent puis modifier ce maillage pour obtenir un maillage de Ω ; c'est l'objet de la section suivante.

2.2.1.2 Mailleur de Delaunay :

Revenons à présent à notre problématique de départ, à savoir : étant donné un maillage de la **frontière de Ω**, comment construire un maillage de Ω ?

Une **méthode de type Delaunay** consiste à :

- **trianguler l'enveloppe convexe des nœuds (Figure 2.11 (a)) en deux étapes :**
  - trianguler une boîte qui englobe tous les nœuds et qui déborde un peu (en 2D il s'agit d'une boite rectangulaire formée de deux triangles) ;
  - **introduire les nœuds *S* les uns après les autres dans la triangulation par le noyau de Delaunay :**
    - en retirant par adjacence tous les éléments dont la boule circonscrite contient ***S*** (**cavité de Delaunay**) ;
    - et en ajoutant les éléments construits avec ***S*** et les faces restantes des éléments supprimés (opération appelée étoilement) ;
- **faire respecter la frontière en effectuant :**
  - des bascules entre arête et arête en 2D ou entre arête et faces en 3D ;

- et en introduisant éventuellement de nouveaux nœuds **(**Figure **2.11 (b)) ;**

- **supprimer les éléments en dehors du domaine (ils peuvent être repérés par coloriage) [52].**

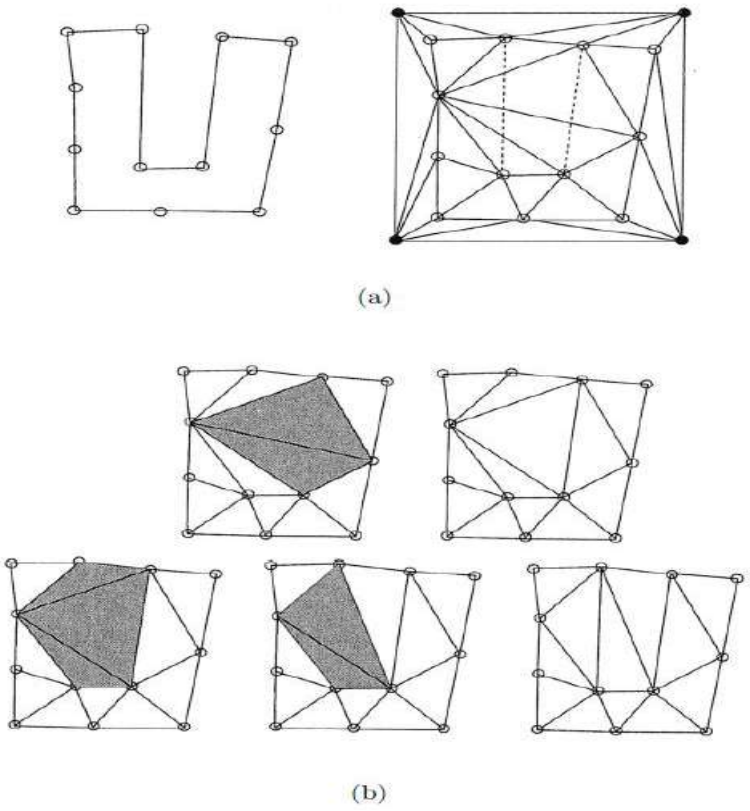


Figure **2.11–** méthode de type Delaunay en 2D, triangulation de l'enveloppe convexe (a) et reconstruction de la frontière (b). [ ]

Le coloriage des éléments en dehors du domaine s'effectue ainsi [53]:

– choisir un sommet de la boîte englobant ;

– tous les éléments dont il est sommet sont extérieurs au domaine ;

– par adjacence, sauf à travers les faces frontières, on récupère tous les éléments de cette composante connexe de l'extérieur ;

– il faut poursuivre récursivement pour récupérer toutes les composantes connexes de l'extérieur.

2.2.1.3 Critique de la méthode :

Ce mailleur rencontre quelques difficultés [29] :

– pendant tout le processus, il faut garder à jour les centres et rayons des boules circonscrites (pour les calculer il faut résoudre un système linéaire ; on peut cependant éviter cette résolution en transportant les centres et rayons connus à l'´etape précédente) ;

– la détermination de **la cavité de Delaunay** par adjacence peut conduire à des cavités non connexes par face, dont les faces frontières ne voient pas toutes *S* ou qui enroulent complètement d'autres nœuds (une étape de correction des cavités est donc nécessaire) ;

– la recherche par adjacence commence par les éléments qui contiennent *S* ; il faut donc les trouver dans le maillage (cf. la méthode de recherche dans le paragraphe suivant) ;

– en 2D, on est sur de pouvoir respecter la frontière (c'est-à-dire d'imposer les faces données au départ) car l'operateur de bascule d'arête permet de visiter toutes les triangulations possibles ; mais en 3D, ce n'est pas le cas et le respect de la frontière peut être difficile, voire impossible à obtenir ; il faut donc bien souvent retravailler longuement le maillage de la frontière avant de pouvoir utiliser le mailleur de Delaunay ;

– il existe un tétraèdre appelé *sliver* (cerf-volant) qui est presque dégénéré (tandis que ni ses arêtes ni ses faces ne le sont) et qui satisfait pourtant le critère de Delaunay (le critère de Delaunay n'est donc pas un critère de qualité) ; en 3D, le mailleur de Delaunay peut faire apparaître de tels éléments indésirables dans le maillage.

La méthode de recherche d'un élément contenant un point *S* (cette opération coûteuse est appelée localisation) est la suivante :

– quand on est dans un élément T qui ne contient pas *S*, *S* voit au moins une face *F* de *T*, on passe alors dans l'autre élément qui partage F, jusqu'à trouver un élément contenant *S* ;

– cela fonctionne si on dispose d'un maillage de l'enveloppe convexe du domaine (car alors il n'y a pas de trou entre *S* et *T*), ce qui doit être le cas dans un mailleur de Delaunay ;

– si *S* voit plusieurs faces *F*, il faut en choisir une de manière aléatoire (sinon, il se peut que le processus boucle indéfiniment) ;

– pour ne pas partir d'un élément initial trop loin de *S*, il est bon d'avoir un arbre d'éléments initiaux (ou même une grille grossière).

2.2.2 Méthodes de type frontales :

Historiquement, les meilleurs frontaux sont les premiers meilleurs automatiques. Ils permettent de construire le recouvrement d'un domaine à partir d'une approximation polygonale de la frontière du domaine. En général ils sont basés sur une technique d'avance de front [54].

Avec cette classe de méthode, le maillage et les nœuds sont générés simultanément contrairement à la méthode de Delaunay qui consiste à connecter un ensemble de points existants [43].

2.2.2.1 Description de la méthode :

Le point de départ de la méthode est une discrétisation de la frontière par une approximation polygonale qui donne une liste d'arêtes en 2D, et une liste de faces triangulaires en 3D [55].

Lors de leur création, ces arêtes et ces faces sont orientées de façon que par convention l'on sache où se trouve le domaine à trianguler (en général, on utilise la règle de la main droite). Ces éléments constituent le front initial et seront la base pour la génération d'un élément du maillage (triangle ou tétraèdre).

On fait progresser le front vers l'intérieur du domaine en ajoutant des sommets intérieurs jusqu'à ce qu'il ne reste plus dans le front que trois arêtes en 2D ou quatre faces en 3D [56].

Exemple d'avance de front :

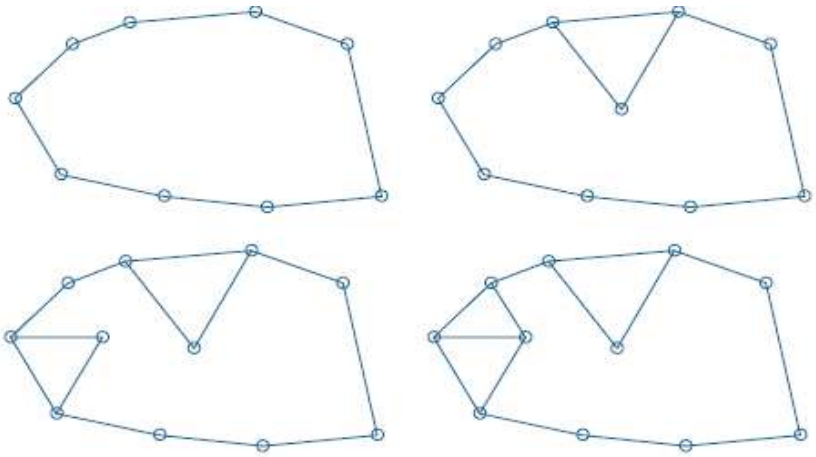


Figure **2.12** a) Quelque étape de front.

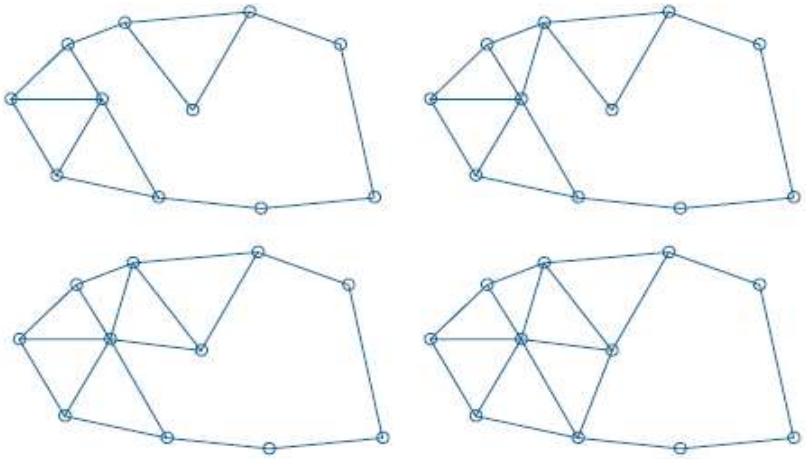


Figure **2.12 b)** Quelque étape de front (suite).

Exemple d'avance de front (fin) :

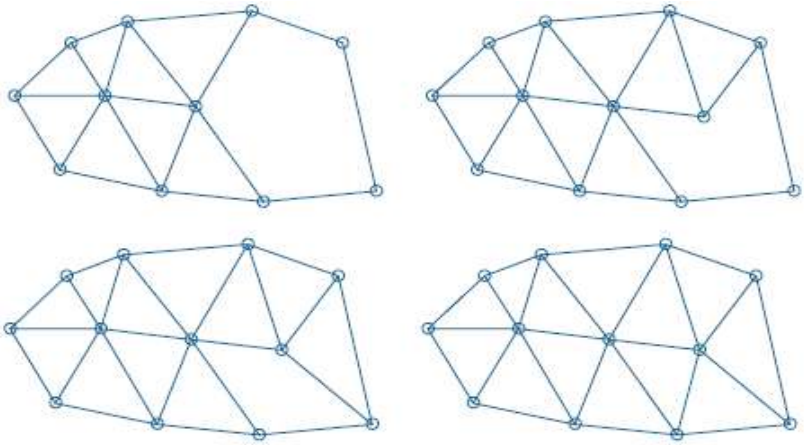


Figure **2.13** Quelque étape de front (fin).

2.2.2.2 Algorithme :

De façon schématique, on choisit un segment ou une face, et on crée un point situé à l'intérieur du domaine à mailler. En reliant ces entités, on obtient un nouvel élément [57].

1. Initialisation du front

2. Analyse du front

   a) Détermination de la zone de départ.
   b) Analyse de cette zone :
      - Création de point(s) interne(s)
      - Construction d'élément(s) interne(s)
      - Mise à jour du front

2. Tant que le front n'est pas vide, aller à 2)

2.2.2.3 Un meilleur frontal :

Le front est une structure dynamique qui évolue au cours du processus. Selon l'algorithme particulier pour la création des éléments, des segments ou des faces sont retirés, et par l'addition de nœuds, on ajoute de nouveaux segments ou faces au front.

Liste d'arêtes	Liste de triangles
S1,S5	S3,S5,S12
S1,S5	S3,S5,S12
….	……

2.2.2.4 Le front initial :

Le front initial est obtenu par la discrétisation des frontières du domaine. La répartition des nœuds sur celles-ci doit répondre à deux critères [59]:

- la représentation de la géométrie.
- la conformité avec la carte des mailles.

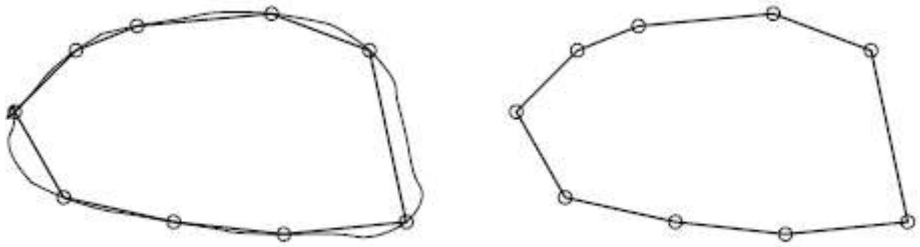


Figure **2.14** Représente la discrétisation des frontières de front.

2.2.2.5 L'analyse du front :

A une étape donnée du processus, le front est constitué de segments actifs c-à-d des segments qui sont susceptibles de servir à la construction d'un triangle. Le choix du segment avec lequel le prochain élément sera construit est **arbitraire**. Une bonne stratégie est de prendre le segment le plus court; ce qui permet une progression ordonnée du front. Ceci est particulièrement important pour les maillages irréguliers [60].

Ensuite on analyse le voisinage du candidat pour déterminer la construction d'un élément. On retient trois situations basées sur l'angle a, entre deux segments successifs.

Cas A): $\alpha < \pi/2$

Cas B): $\pi/2 < \alpha < 2\pi/3$

Cas C): $2\pi/3 < \alpha$

2.2.2.5.1 Cas de figure A [61]:

Soit α, l'angle entre deux arêtes consécutives du front. Si $\alpha < \pi/2$, alors les deux arêtes sont retenues et avec le segment S_2S_4, elles vont former le triangle créé.

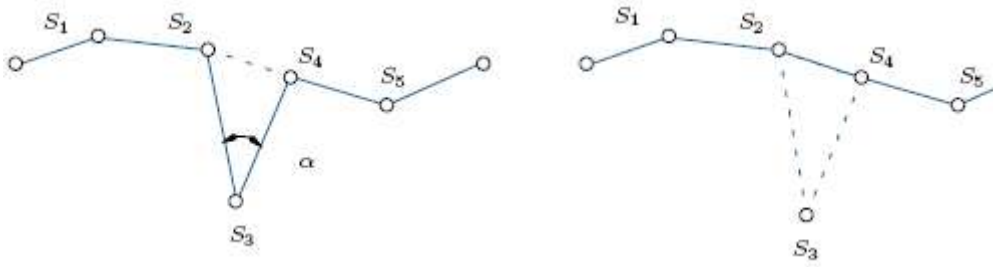


Figure **2.15** Cas A) et construction associée

Le front est mis à jour en retirant les arêtes S_2S_3 et S_3S_4, et en y ajoutant le segment S_2S_2.

On note que dans ce cas de figure la création d'un élément se fait sans ajout de sommets.

2.2.2.5.2 Cas de figure B [56]:

Soit α, l'angle entre deux arêtes consécutives du front. Si $\pi/2 < \alpha < 2\pi/3$, alors à partir de ces deux arêtes, on génère un sommet interne et deux triangles.

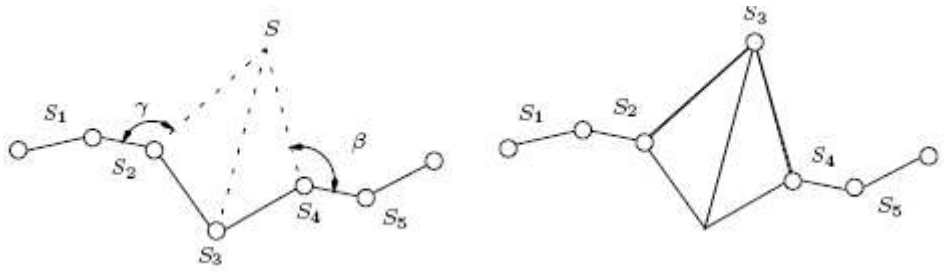


Figure **2.16** Cas B) et construction associée

Le sommet S est généré sur la bissectrice de l'angle à une distance calculée à partir des longueurs respectives des arêtes de la zone de départ. On utilise une moyenne pondérée de la forme :

$$d_{SS_3} = \left(d_{S_1S_2} + 2d_{S_2S_3} + 2d_{S_3S_4} + d_{S_4S_5}\right)/6 \qquad 2.20$$

2.2.2.5.3 Cas de figure C [57]:

Soit α, l'angle entre deux arêtes consécutives du front. Si $2\pi/3 < \alpha$, alors une seule arête est retenue et un triangle est construit avec cette arête et un sommet interne.

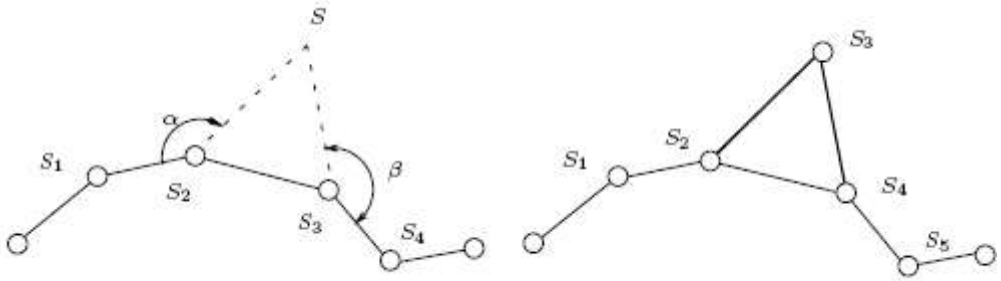


Figure **2.17** Cas C) et construction associée

Un triangle aussi équilatéral que possible est formé à partir de l'arête de la zone de départ ayant la longueur minimale.

2.2.2.6 Création du sommet :

La position du sommet dépendra, en l'absence d'une carte de maille, sur les segments voisins. Deux constructions possibles sont illustrées (Figure **2.18**) **[55]**:

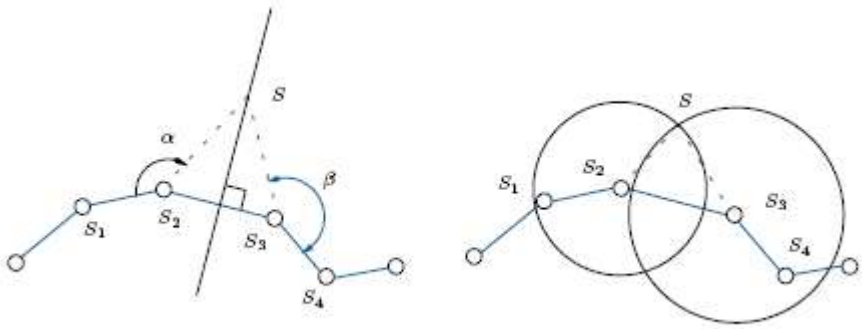


Figure **2.18** les Deux constructions possibles pour la création du sommet

2.2.2.7 Position de sommet cas1:

Le sommet est localisé sur la bissectrice perpendiculaire à une distance :

$$d_s = \frac{w_1 d_{s_1 s_2} + w_2 d_{s_2 s_3} + w_3 d_{s_3 s_3}}{\sum_{i=1}^{i=3} w_i} \qquad 2.21$$

où les poids w_i sont appliqués aux segments d_j. Les valeurs typiques sont $w_l = 1$, $w_2 = 2$ et $w_3 = 1$[52].

2.2.2.8 Position de sommet cas2:

Le sommet est localisé à l'intersection de deux cercles centrés sur les sommets S2 et S3, de rayons, respectivement [58]:

$$d_{ss_2} = \frac{\left(w_1 d_{s_1 s_2} + w_2 d_{s_2 s_3}\right)}{\sum_{i=1}^{i=2} w_i} \qquad 2.22$$

$$d_{ss_3} = \frac{\left(w_2 d_{s_2 s_3} + w_3 d_{s_3 s_4}\right)}{\sum_{i=1}^{i=3} w_i} \qquad 2.23$$

Où les poids w_i sont appliqués aux segments avec des valeurs typiques de $w_1 = 1$, $w_2 = 2$ et $w_3 = 1$.

Ces paramètres sont arbitraires.

Finalement, à chaque fois qu'on analyse le front, on essaye d'abord d'appliquer le cas A.

- Si le cas A ne peut s'appliquer en aucun endroit du front, alors on essaye ensuite d'appliquer le cas B.
- Si le cas B ne peut s'appliquer en aucun endroit du front, alors on essaye ensuite d'appliquer le cas C.
- S'il y a plusieurs parties du front où on peut appliquer un cas en particulier, il semble qu'il soit préférable de commencer par les parties les plus petites.

2.2.2.9 Coût d'un meilleur frontal :

La plupart du temps de calcul d'un meilleur frontal est perdu dans les deux problèmes suivants [60]:

- Étant donné un sommet S qu'on voudrait générer, chercher s'il n'y aurait pas un sommet du front dans le voisinage de S.

- Étant donné une arête A qu'on voudrait générer, chercher si elle n'intersecté pas une autre arête du front.

2.2.2.10 Optimisation d'un mailleur frontal [61]:

Pour optimiser cette recherche, il suffit pour un sommet S ou pour une arête A donné, d'avoir la liste des arêtes du front qui sont dans les alentours et de ne parcourir que cette liste.

Cette classification est très sensible à la tolérance numérique.

On peut utiliser un recouvrement du domaine avec une grille régulière. Dans chaque case de la grille, on stock les arêtes du front qui sont dans la case et on tient cette liste à jour.

Dans la même veine, on peut utiliser un arbre binaire. Les arbres digitaux alternés seraient la meilleure des stratégies.

Dans ce cas, il semblerait que les meilleurs frontaux seraient d'ordre O(n1).

2.2.2.11 Critique de la méthode [41]:

Les principales difficultés rencontrées par cette méthode sont les suivantes :

- par où commencer parmi les faces du front ?
- comment choisir le nœud à introduire ?
- comment savoir si un élément intersecté le maillage ?

Concernant le choix de la première face du front à traiter, on peut prendre la plus petite face ou celle de plus mauvaise qualité. Concernant la position du nœud à introduire, les candidats sont dans l'ordre décroissant de priorité : des nœuds proches, le point optimal (celui qui permet de construire le simplexe de meilleure qualité possible) et enfin, des candidats donnés a priori (avant le début du processus). Concernant, la phase de détection de collision des fronts, qui est généralement coûteuse, il est nécessaire d'employer une méthode efficace de localisation.

La méthode frontale reste la meilleure technique pour produire le plus d'éléments de bonne qualité.

Cependant, les inconvénients de cette méthode restent tout de même les suivants : le front peut devenir **chaotique** (même pour des géométries simples, surtout en 3D) et tous les points candidats peuvent être rejetés et l'algorithme ne pas converger (sauf en 2D), il faut alors supprimer une partie du voisinage et tout recommencer**. Il y a donc un problème général de convergence avec la méthode frontale.**

2.2.3 Méthodes de quadtree-octree (méthode par arbre):

Introduite il y a maintenant une vingtaine d'années (M.A. Yerry and M.S. Shephard (1984) [42]), la méthode par arbre (ou décomposition spatiale) est aujourd'hui activement utilisée en 3D (SHARIFI, H, (1996) [62]). Une des caractéristiques principales de ces méthodes, est de générer le maillage de la frontière du domaine considère en même temps que le maillage de l'intérieur de ce domaine, en utilisant la description CAO de cet objet.

Soit une cellule mère (un rectangle en 2D, un parallélépipède en 3D) englobant l'objet à mailler. La première étape consiste à décomposer de manière récursive chaque cellule contenant plusieurs nœuds en 2d cellules de mêmes tailles (Figure **2.19**(b)). Un critère d'arrêt simple pouvant être : chaque cellule contient au plus un nœud. On obtient alors un arbre 2d-aire appelé quadtree en 2D et octree en 3D **[26].**

La technique consiste ensuite à :

- supprimer les cellules extérieures au domaine ;

- trianguler les cellules contenant un nœud, en respectant la frontière donnée ;

- éliminer, dans ces cellules, les éléments extérieurs au domaine ;

- trianguler les cellules vides en considérant leurs sommets comme des nœuds supplémentaires (Figure **2.19**(c)).

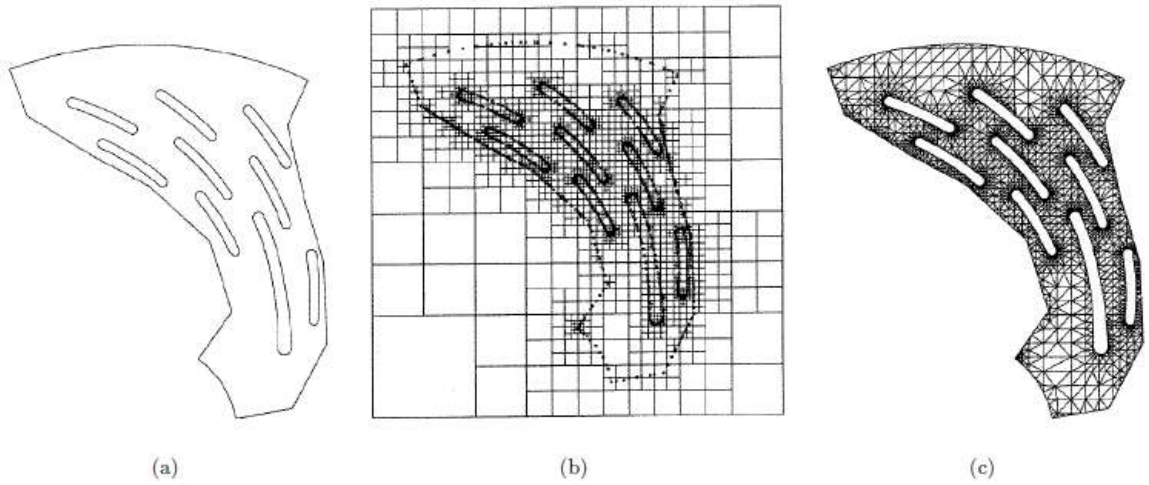


Figure **2.19** – Méthode par arbre en 2D.[62]

2.2.3.1 Critique de la technique [34] :

Avec cette méthode, on peut se retrouver **face aux problèmes suivants** :

- un nœud de la frontière peut se trouver au bord, voire au coin d'une cellule, ce qui peut conduire à des éléments **de mauvaise qualité ;**

- le critère d'arrêt précédent est trop pauvre car on peut se retrouver avec la configuration suivante : une cellule contient un nœud et la portion d'une autre face ;

- de nombreux nœuds inutiles sont introduits en frontière ;

- pour éviter que l'arbre soit mal équilibré et que les tailles de cellules voisines soient très différentes, on peut équilibrer l'arbre en forçant la règle 2 : 1 (i.e. la différence de niveau entre deux cellules voisines est limitée à un facteur 2), ce qui introduit énormément de nœuds dans le volume.

L'avantage principal de cette technique est **la structure d'arbre** qui **est très utile** (pour faire des localisations dans **le maillage** par exemple). Son inconvénient majeur est la quasi impossibilité de générer des éléments isotropes.

2.2.4 Autres méthodes :

Après ces méthodes devenues classiques dans leur formulation générale, nous allons nous intéresser à quelques approches moins courantes pour la génération de maillages non

structures, L'objectif n'est pas de les analyser en détail, mais simplement de montrer la diversité des techniques de maillage.

2.2.4.1 Méthode de « paving » :

Exposée dans (Blacker,Stephenson-(1991)[63]),cette méthode consiste à décomposer en blocs un domaine en fonction de sa géométrie, puis à mailler chaque bloc de géométrie simple en hexaèdres via des méthodes de type algébrique (*submapping*). **C'est donc une méthode de blocs produisant au final des maillages généralement non structure**s, *Le paving* crée des éléments hexaédriques de bonne qualité. En revanche, cette méthode s'applique difficilement aux objets de géométries **très complexes**. La Figure **2.20**, extraite de (Blacker-(1996) [64]) présente un exemple de maillage réalisé à laide de « l'outil du tonnelier », dérivé de la méthode de *paving.*

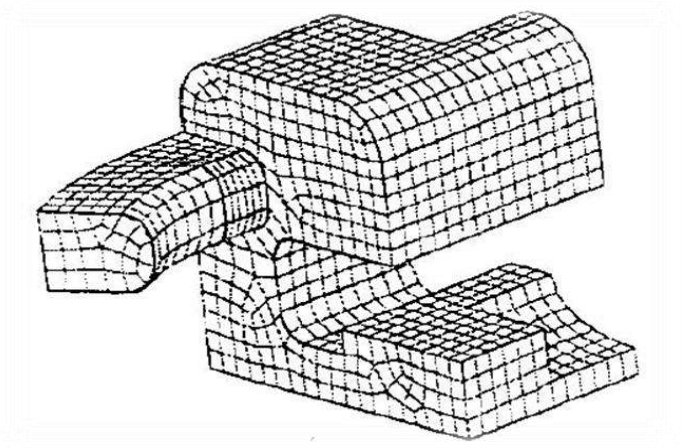


Figure **2.20** -Maillage par la méthode « du tonnelier » [63]

2.2.4.2 Méthode de squelette :

Cette méthode, développée dans (Price et al.1995 [65]) et (Price,Armstrong (1997) [66]) repose sur la notion mathématique de squelette. Le squelette d'un objet est approche par une « ligne » médiane en dimension deux qui est utilisée pour découper cet objet en blocs. Chaque bloc est alors maille en quadrangles, par exemple par une méthode algébrique, en veillant à la conformité du maillage global. La Figure **2.21** illustre une méthode de ce type appliquée au cas très simple d'un rectangle.

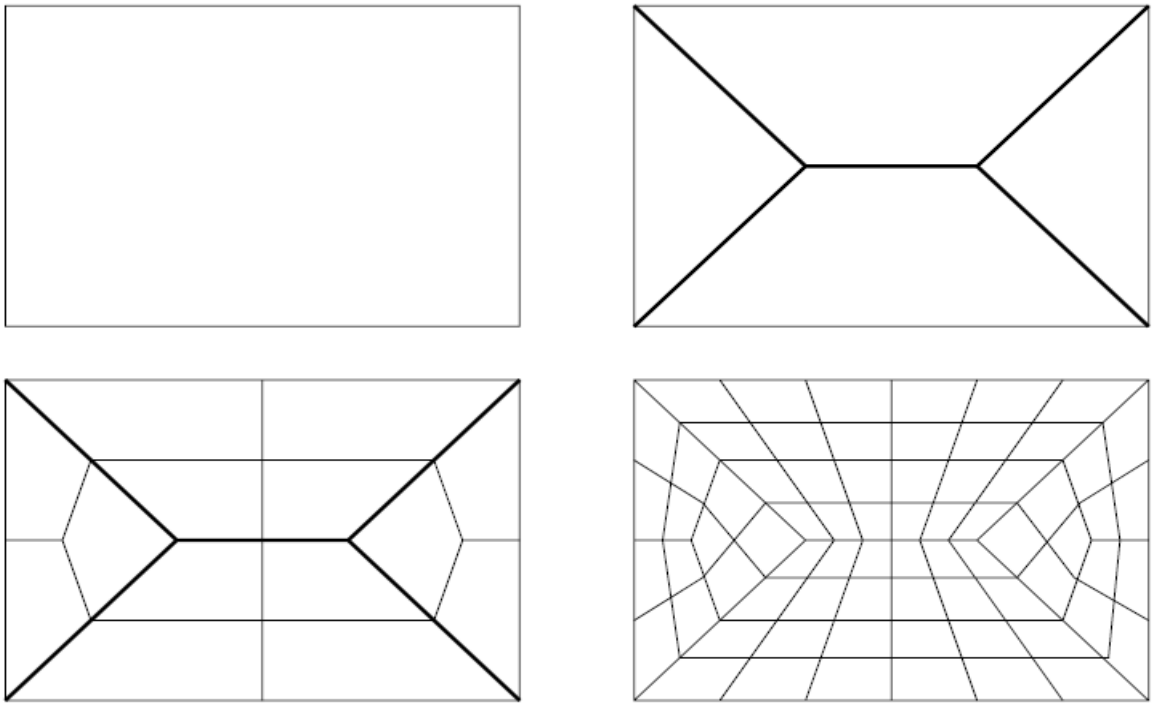


Figure **2.21**- Maillage d'un rectangle par une méthode de squelette. En haut à droite, le squelette du rectangle apparait en gras. Les blocs ainsi formes sont découpés pour n'avoir que des sous quadrangles (en bas à gauche). En bas à droite, le maillage final (non régularisé) est obtenu par l'application d'une méthode algébrique à chaque sous-quadrangle. [65]

2.2.4.3 Méthode topologique :

Proposée par (Coupez-(1991) [1]), cette méthode s'applique en dimension deux, ou en dimension trois par exemple pour le maillage de pièces mécaniques obtenues par forgeage. En dimension trois, un maillage tétraédrique valide de l'objet initial est généré. Cet objet subit une déformation mécanique. Tout en conservant la même topologie, le maillage initial est alors transformé géométriquement, par modification des coordonnées des points. De ce fait, il devient faux la plupart du temps : certains éléments ont un volume négatif. Par des transformations topologiques et l'ajout éventuel de points internes, ce maillage est alors corrigé de manière à le rendre valide à nouveau. Le procédé peut être itéré autant que nécessaire en suivant la déformation de la pièce. Les Figures **2.22** et **2.23** illustrent une méthode de ce type appliquée en dimension deux sur un exemple académique.

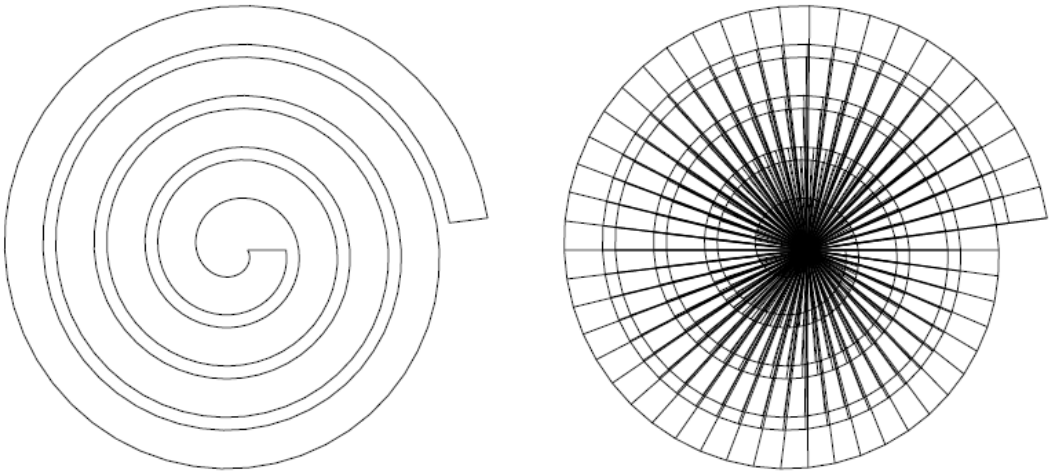


Figure **2.22** - Maillage par une méthode topologique. À gauche, la donnée initiale. À droite la première étape consiste à connecter toutes les arêtes de la donnée au point le plus proche du barycentre de l'objet, Le maillage obtenu est typologiquement valide mais géométriquement faux. [1]

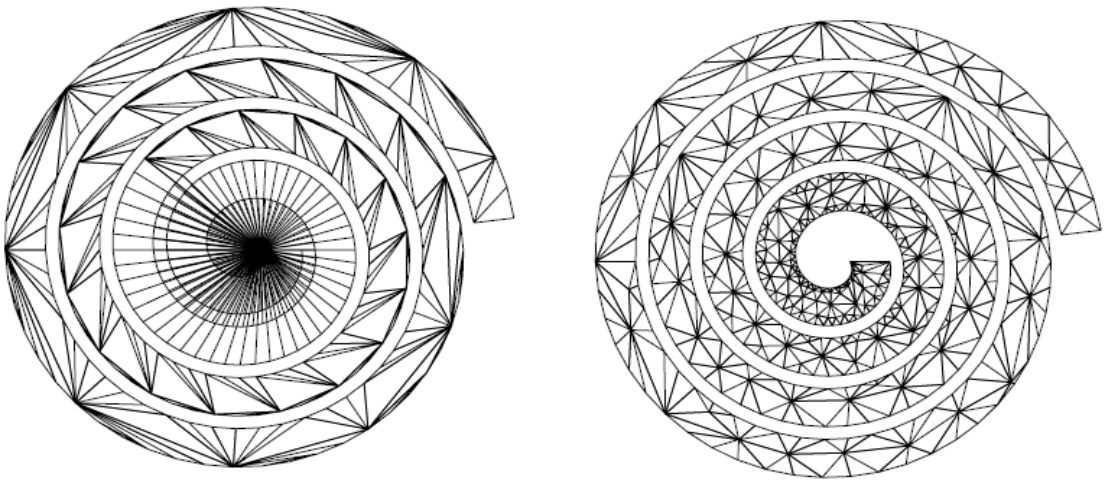


Figure **2.23** - Maillage par une méthode topologique (suite et fin). À gauche, une étape intermédiaire du processus de maillage. Par des transformations topologiques autour de chaque point, le maillage devient géométriquement valide. À droite, le maillage final est correct mais pas optimisé. [1]

2.3 Quelques problèmes fréquents [34]:

Malgré l'existence de nombreux algorithmes, des avancées mathématiques et informatiques en terme de programmation et d'algorithme, le problème de la génération d'un bon maillage est loin d'être résolu. En effet les différentes demandes des utilisateurs engendrent de nombreuses difficultés qui nécessitent le savoir-faire et la patience du spécialiste du maillage.

Afin de faciliter la tâche il est indispensable de mieux prendre en compte les besoins des utilisateurs comme :

- Le transport efficace de sommets d'une zone à une autre dans le maillage pour avoir une discrétisation suffisante aux endroits stratégiques.

- L'obtention d'un maillage correct après modification de la géométrie de l'objet considéré.

- Les problèmes de la représentation rigoureuse de la frontière de domaines complexes. La complexité des problèmes abordés, notamment en 3D rend l'utilisation et la maîtrise des algorithmes proposés dans la littérature fastidieuse et difficile.

Conclusion :

Cette étude bibliographique montre qu'il existe plusieurs algorithmes pour la génération automatique de maillage.

Dans notre étude on a adopté la méthode par blocs selon l'algorithme présenté dans les travaux [3,15] pour la possibilité d'amélioration et introduction des modifications

3. Méthodes de blocs :

3.1 Méthode de blocs en dimension 2 :

Les méthodes de blocs, parfois connues sous le nom de méthodes de superéléments, ont été introduites et étudiées en particulier par (Thompson-(1987) [4]).

L'approche par bloc, en particulier dans le cas bidimensionnel, représente une solution permettant de créer le maillage de domaines géométriquement complexes pour lesquels les méthodes précédemment étudiées ne sont pas adaptées.

Une autre application prometteuse réside dans la possibilité de résoudre en parallèle le problème traité sur chacun des blocs.

L'idée de base consiste à découper le domaine considéré en une série de blocs de topologies élémentaires (triangles, quadrangles). Chaque bloc représente un élément de cette partition grossière. Le maillage du domaine complet est obtenu par recollement des différents maillages des blocs de la partition.

La technique de blocs induit deux types de difficultés concernant :

• la création des blocs élémentaires et leur maillage, du choix de ces blocs dépendra la nature du maillage final ;

• la gestion des interfaces entre les blocs de façon à permettre le recollement correct de leurs maillages.

3.1.1 Algorithme de maillage :

Adapté aux géométries à priori quelconques, ce type de générateur va construire le recouvrement du domaine en segments, triangles et quadrangles à partir d'un maillage grossier de ce domaine composé de ces mêmes éléments et de **points de contrôle** définis sur les arêtes de ces **éléments grossiers** ou blocs. Ces points sont utilisés pour découper chaque bloc en sous-éléments du même type [28].

La méthode comprend quatre étapes :

Etape 1 : La construction du recouvrement grossier et la définition des points de contrôle, c'est-a-dire :

•la définition des sommets des blocs ;

•l'énumération de chaque bloc en termes de sommets ;

•la définition des ***points de contrôle*** sur les arêtes des blocs :
Numéro de points et position de ceux-ci; on choisira un même nombre de points de contrôle sur deux arêtes logiquement connectées [33].

Etape 2 : Le découpage de chaque bloc en fonction de la position et du nombre de points sur ses arêtes ; cette opération repose sur :

•le report sur l'élément de référence, de même nature géométrique que le bloc traité, des points de contrôle ;

•le découpage canonique de cet élément de référence en utilisant ces points ;

•le transport du maillage précédent sur le bloc réel.

La difficulté réside dans l'énumération correcte des sommets des éléments c'est-a-dire dans la construction de la connectivite **(c'est l'objet de l'étape 3) [62].**

Etape 3 : L'énumération des éléments issus du découpage : pour obtenir les éléments composant le maillage du bloc, il suffit d'énumérer leurs sommets.

Pour ce faire, on utilise une numérotation formelle en *i,j* puis on trouve le numéro global des sommets via un pointeur ***P(i,j)*** construit à cet effet.

Afin de préciser ce processus, nous prenons le cas d'un bloc de nature quadrangulaire. On désigne par les couples (0,0), (n1, 0), (n1, n2) et (0, n2) les numéros formels des 4 sommets grossiers du bloc (n1 et n2 représentent le nombre de points - 1 (ou le nombre de subdivisions) sur les arêtes 1 et 3 de ce bloc d'une part et 2 et 4 d'autre part):

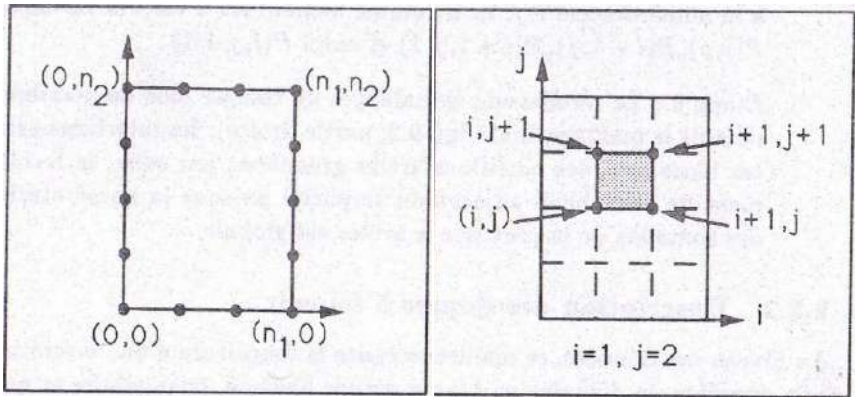


Figure 3.1 : Numérotation formelle.

Les indices i et j d'un sommet quelconque repéré dans l'espace défini par les 4 sommets permettent d'associer a ce sommet le couple (i,j).

i,j ⟹ (i,j)

Il est trivial de voir qu'un élément du découpage (figure 3.1, partie droite) est connu par ses 4 sommets de couples (i,j),(i + 1, j), (i+1, j +1) et (i, j +1).

Cette énumération formelle permet de calculer les numéros globaux des sommets :

(i,j) ⟹ N = P(i,j)

Où le pointeur P est construit de la façon suivante :

•si le sommet S est un sommet du maillage grossier, P (i, j) est son numéro dans ce maillage ;

•si le sommet S est sur une arête grossière, P (i,j) est calcule grâce au numéro de cette arête dans le maillage grossier;

•pour S interne, F (i,j) est défini séquentiellement a partir du premier numéro libre (i.e. après le dernier numéro du dernier point de la dernière arête grossière).

Connaissant P, on peut alors trouver les 4 sommets de tout élément a partir des 4 couples (i,j),(i+1,j), (i+1,j+1) et (i,j + 1) associes a la numérotation i,j: ils ont comme numéro les 4 valeurs suivantes: P(i,j),P(i +1,j),P(i +1, j 1) et enfin P(i,j 1) [10].

Etape 4 : Le recollement des maillages de chaque bloc de manière à obtenir le maillage final : les interfaces entre ces blocs sont des partitions d'arêtes grossières ; par suite, le recollement de deux blocs adjacents est implicite puisque la numérotation des sommets de la partition des arêtes est globale.

D'autre part, il faut veiller à ce que le maillage global, obtenu après assemblage des maillages des différents blocs, soit toujours correct, essentiellement aux frontières entre blocs. Notons par ailleurs que ces méthodes autorisent l'utilisation de différents types d'éléments pour mailler les blocs, à condition que la connectivité reste correcte. On obtient alors un maillage mixte [34].

3.1.2 Mailleur à super éléments :

La technique de maillage utilisant les interpolations isoparamétriques fut l'une des premières à être utilisées pour les géométries complexes. Elle permet de raffiner la modélisation d'un domaine discrétisé par des super éléments à 4 ou 8 nœuds [67]. Ces derniers (8 nœuds) permettent de raffiner le maillage des contours courbes.

Le raffinement dépend des nombres de subdivisions fixées pour les super éléments.

Le modélisateur doit respecter la compatibilité des subdivisions sur les frontières communes aux super éléments.

Le modèle de la Figure 3.2 est obtenu par le mailleur super éléments et l'option de jonction de modèles symétriques [29].

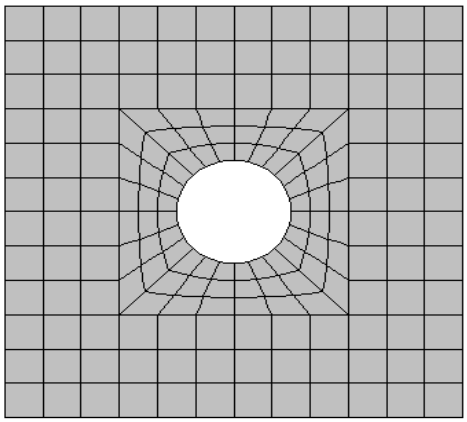


Figure 3.2: Plaque trouée

La forme des blocs dans l'espace cartésien (géométrique) est choisie de façon qu'elle puisse être construite par une transformation isoparamétrique standard d'une forme simple dans l'espace de référence mathématique (ξ et η en 2D) [68-69].

L'espace de référence mathématique est souvent appelé plan parent en 2D et les formes simples s'appellent éléments de référence et sont normalement, des triangles ou des rectangles dans l'espace à deux dimensions et des cubes ou hexaèdres dans l'espace en trois dimensions. La figure (3.3) montre la décomposition d'un objet en super-éléments.

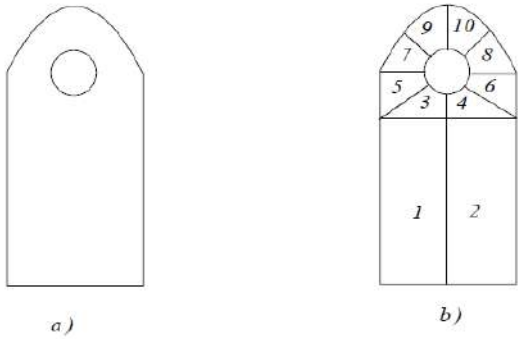


Figure 3.3 : a) l'objet à mailler b) décomposition de l'objet en 10 superéléments

La figure (3.4) présente les transformations d'un super-élément de l'espace réel (x, y) vers l'espace (ξ, η) et l'inverse. La génération du maillage se fait dans l'espace de référence (ξ, η) et le maillage généré est ensuite transféré à l'espace réel (x, y).

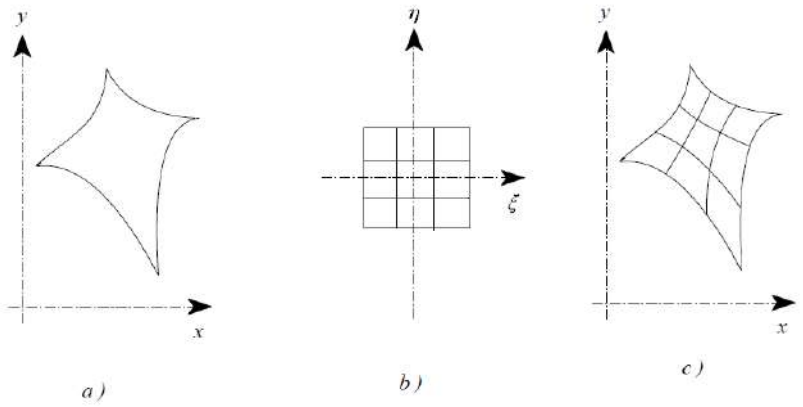


Figure 3.4: a) super-élément dans l'espace x et y. b) transformation du superélément dans l'espace de référence ξ et η et le maillage généré. c) transformation du maillage en l'espace x et y.

La raison pour laquelle on utilise la transformation isoparamétrique est que le maillage d'un objet dans l'espace paramétrique est plus simple à réaliser que dans l'espace réel. Dans l'espace paramétrique, nous sommes en présence de formes simples, comme des triangles ou des rectangles en deux dimensions, ou des hexaèdres en trois dimensions, que l'on peut mailler très facilement [19].

Par exemple, en 2D, si les coordonnées des sommets de l'objet dans l'espace réel (x_i et y_i) sont connues, on peut calculer facilement les coordonnées dans l'espace **réel de <u>i</u> n'importe** quel point de l'objet en utilisant les équations suivantes:

$$x(\xi,\eta) = \sum_{i=1}^{\text{nb-noeuds}} N_i(\xi,\eta)\, x_i \quad , \quad y(\xi,\eta) = \sum_{i=1}^{\text{nb-noeuds}} N_i(\xi,\eta)\, y_i \qquad 3.1$$

Les $\mathbf{N_i}$ sont les fonctions de forme ("shape function") et (ξ, η) sont les coordonnées paramétriques dans le plan parent ou le cube de référence. nb-nœuds représentent le nombre de nœuds par élément [39].

Parce que le super-élément, par exemple, en 2D est divisé selon les lignes ξ constant et η constant, le nombre de nœuds (ou de division) sur les côtés opposés doit être le même. Cette situation crée quelques restrictions pour la génération de maillage. Dans le cas où les tailles des côtés opposés d'un super-élément seraient très différentes, le maillage généré n'est pas uniforme. Les éléments proches des côtés longs sont plus grands que les éléments proches des petits côtés [70].

4. Programme informatique :

4.1. Introduction:

Le code générateur de maillage, développé au cours de ce travail et intitulé MESH_GENERATOR_2D, est écrit en "FORTRAN 77". La machine sur laquelle il a été développé, est un micro-portable HP CQ 61.

Ce programme développé au cour de ce travail est version amélioré et modifié du programme cité dans la référence [3] ou quelque routines sont donnés est complétés dans les références [15,17].on introduit plusieurs modifications dans ces routines afin de les mètres en marcher d'une part et donner une cohérence au programme globale d'une autre part.

En autre, la bibliothèque de ce programme a été enrichie par l'introduction d'une variété de types éléments à savoir, les éléments linéaires à 3ou4 nœuds, les éléments quadratiques 6,8et 9 nœuds.

Le préprocesseur ici développé qui peut être utilisé pour la génération automatique de maillage dans l'analyse par éléments finis des structures en 2D.

En effet, l'analyse d'une structure par la méthode des éléments finis suit trois étapes principales:

a) Le Pré-traitement:

C'est la phase de préparation des données ; en l'occurrence :

Données géométriques, propriétés des matériaux et conditions de charge et d'appui.

b) Le traitement :

C'est la phase de calcul des matrices élémentaires et globales ainsi que la résolution du système d'équations obtenu.

c) Le Post-traitement :

C'est l'étape de présentation des résultats.

La préparation des données est une étape très importante puisqu'elle représente une source d'erreurs difficiles à repérer et longues à réaliser surtout si la discrétisation se fait manuellement.

Un générateur automatique de maillage devient donc nécessaire pour minimiser le risque d'erreur à la phase de préparation des données.

D'autre part, l'utilisation des éléments triangulaires lors du maillage permet d'épouser les formes complexes des structures.

4.2. Génération d'un maillage automatique [22, 39] **:**

Les étapes de base à suivre pour la génération d'un maillage d'une structure en 2D sont les suivantes :

4.2.1Division de la structure en blocs :

Les zones avec matériaux ou chargements différents sont représentées par des blocs (super-éléments) séparés. Chaque bloc (super-élément) a une forme générale d'un quadrilatère à cotés curvilignes. Il est représenté par un élément isoparamétrique à huit nœuds. Comme données d'entrée, on doit spécifier la topologie et les coordonnées des huit points représentatifs de chaque bloc.

Soit (ξ,η) un système de coordonnées naturelles. Les coordonnées x (ξ,η) et y (ξ,η) d'un point quelconque (ξ,η) à l'intérieur du bloc sont données par les relations suivantes; Figure 4.1 :

$$\begin{cases} x(\xi,\eta) = \sum_{i=1}^{8} N_i(\xi,\eta).x_i \\ y(\xi,\eta) = \sum_{i=1}^{8} N_i(\xi,\eta).y_i \end{cases} \qquad (4.1)$$

Où: xi et yi sont les coordonnées des points définissant le contour du bloc quadrilatéral

Ni(ξ,η) sont les fonctions de forme pour un élément quadratique. Elles sont données par les expressions suivantes :

*pour les nœuds des angles (i=1, 3, 5, 7):

$$N_i(\xi,\eta) = \frac{1}{4}(1+\xi.\xi_i).(1+\eta.\eta_i).(1+\xi.\xi_i+\eta.\eta_i) \qquad (4.2)$$

*pour les nœuds du milieu (i=2, 4, 6, 8):

$$N_i(\xi,\eta) = \frac{1}{2}(1+\xi.\xi_i).\xi_i^2.(1-\eta^2) + \frac{1}{2}(1+\eta.\eta_i).\eta_i^2.(1-\xi^2) \qquad (4.3)$$

Les fonctions de forme ont les propriétés suivantes:

* Elles satisfont les conditions :

$$\sum_{i=1}^{8} N_i(\xi,\eta) = 1 \qquad (4.4)$$

$$N_i(\xi_i,\eta_i) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \qquad (4.5)$$

* Elles deviennent quadratiques le long des cotés de l'élément:

$$\xi = \pm 1 \quad , \quad \eta = \pm 1$$

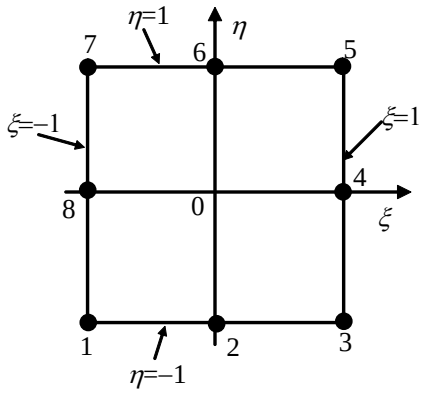


Figure 4.1 élément quadratique isoparamétrique (8 nœud)

4.2.2 Subdivision de chaque bloc en éléments:

Pour chaque super-élément, on doit spécifier le nombre de subdivisons à faire dans chaque direction ξ et η.

4.2.3 Connections des blocs:

Après subdivision de chaque bloc et numérotation de ses points nodaux séparément, il sera nécessaire de connecter tous les blocs et unifier la numérotation des nœuds tout en évitant la répétition pour les points ayant même coordonnées. Pour cela, on compare les coordonnées du nœud 1 avec tous les autres nœuds, le nœud 2 avec tous les autres nœuds et ainsi de suite jusqu'à ce que tous les nœuds répétés soient éliminés.

4.3 Description du programme MESH_2D:

Le programme **MESH_2D** développé dans ce sens comporte :

- un programme principal,

- sept sous-programmes.

L'organigramme du programme général est donné par la figure (4.2)

4. 3.1 Introduction des données et sortie des résultats:

L'introduction des données se fait par un fichier MAIL.DAT. Il comporte:

* Les caractéristiques géométriques de la structure: dimensions, nombre de blocs, coordonnées des points définissants chaque bloc, connectivités chaque bloc (superélément).

* Les caractéristiques physiques: matériau de chaque bloc (superélément).

Le fichier des résultats OUTPUT.DAT comporte:

* les caractéristiques géométriques de la structure discrétisée : dimension

* les caractéristiques du maillage:

  - coordonnées des nœuds.

  - connectivités des éléments générés.

  - matériau de chaque élément généré.

4.3. 1.1 Description le fichier de données :

Des instructions détaillées sur **l'introduction des données** sont données ci-dessous.

1-Le titre de fichier de données est spécifié dans la ligne une.

2-Les sept numéros dans la deuxième ligne sont :

- nombre de régions d'entrée (super éléments) ;

- nombre de nœuds d'entrée ;

- nombre de systèmes de coordonnée polaires auxiliaires,

- type de maillage généré (0 = maillage d'ordre uniforme),

- décision de la température :

(0 = n'entrent pas les températures ; 1 = entrées les températures),

- décision d'épaisseur :

(0 = n'entrent pas l'épaisseur ; 1 = entrée l'épaisseur),

- décision de la renumérotation :

(0 = ne pas renuméroter les nœuds généré ; 1 = renuméroter les nœuds générés).

3- Le nombre de système de coordonnées polaires utilisé pour les entrées des données et de leurs décalages x et y par rapport aux axes cartésiens globales sont ensuite entrés.

L'utilisation des systèmes polaires auxiliaires pour l'introduction des données a été claire.

Pour les super-éléments, seulement un système polaire est nécessaire pour spécifier les coordonnées du semi cercle. Les décalages de x et de y de ce système polaire sont les deux égale 0.

Le système de coordonnées :

(**0** = système principal cartésienne ; **1** = numéro de référence du système polaire 1, etc.), on spécifie le numéro de nœud, et les coordonnées x et y ou le rayon et l'angle (en degrés) des nœuds d'entrée.

4- Le numéro de l'élément, le code de matériau et d'autres informations des super-éléments sont précisées dans les dernières lignes de l'entrée de données.

Dans chaque ligne, le nombre du super-élément est introduit d'abord, suivi par le nombre de subdivisions dans les directions **ξ** et **η**. suivi par indicateur des nœuds de frontière

de chaque d'entrée de super-éléments.

4. 3.2 Description les sous programmes de mailleur (MESH_2D) :

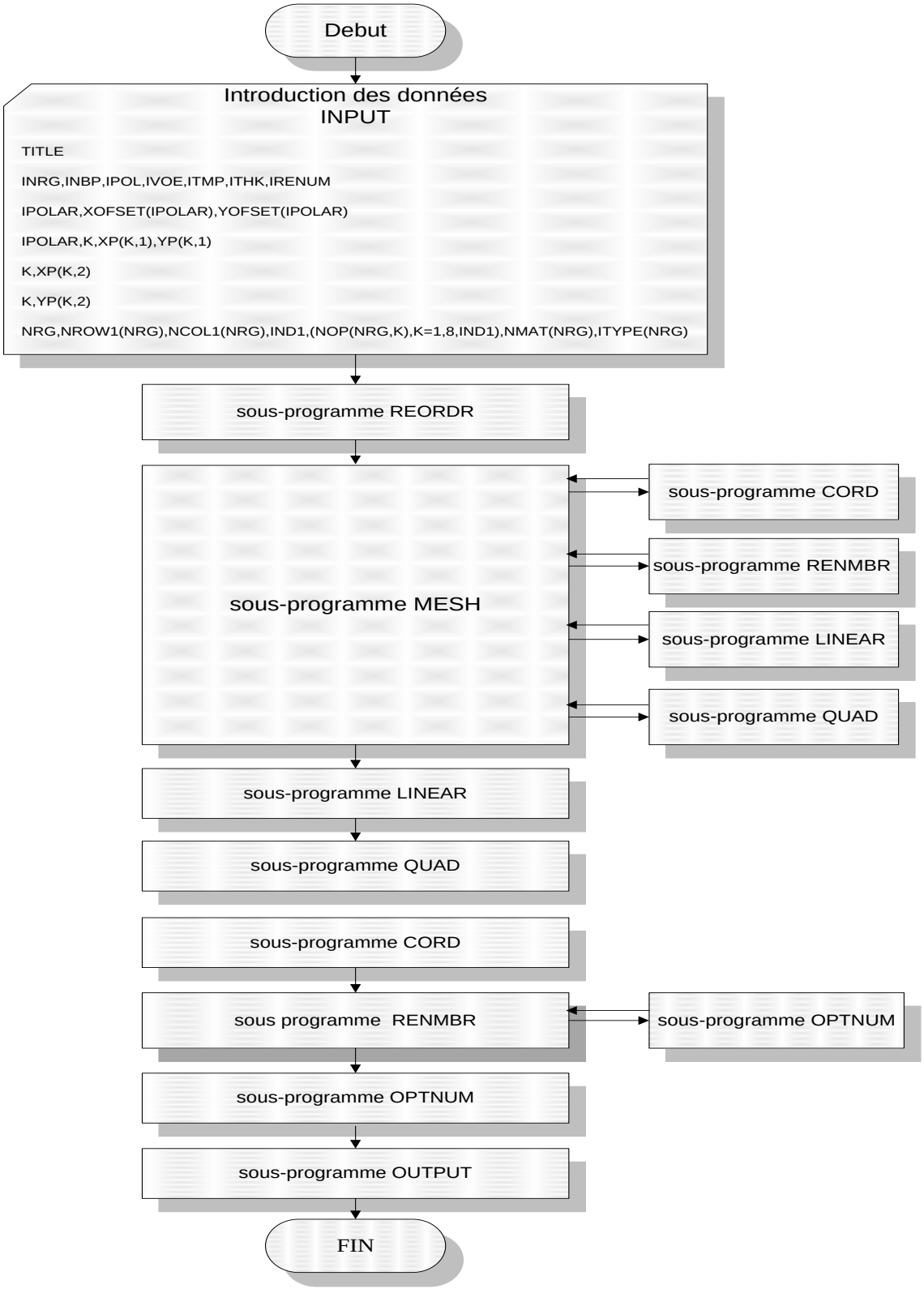


Figure (4.2) représente l'organigramme du programme général

Le programme de Fortran (MESH_2D.for) est à peu près évident, L'organigramme du programme est illustré à la figure (4.2). Il s'agit d'un programme principal et de sept sous-programmes. Cette forme modulaire devrait permettre aux utilisateurs facilement de remplacer ou améliorées les sous programmes ou bien d'effacer les sous programmes qui ne sont pas nécessaires pour leur application, par exemple, la renumérotation des nœuds.

Les définitions des variables utilisées dans ce programme peuvent être trouvées au début du programme principal.

Des descriptions des différentes sous programmes sont données ci-dessous.

4. 3.2.1 Le sous-programme INPUT:

Le sous-programme **INPUT** d'après la figure (4.2), lit les données nécessaires pour définir la géométrie de l'objet, les propriétés des matériaux et le type de maillage et l'ordre des éléments de maillage qui seront générer dans chaque super-élément. Le sous programme **INPUT** vérifie également les données d'entrée et les données de sortie.

4.3.2.2 Le sous-programme REORDR :

Le sous-programme REORDR renumérote les super-éléments de telle manière que, pendant que les nœuds sont générés pour chaque super-élément aucun nombre de nœud double ne soit créé, L'organigramme du sous-programme est illustré à la figure (4.3).

Dans certains cas, il est nécessaire de renuméroter les nœuds générés afin de réduire la largeur de bande. Si la renumérotation des nœuds générés est souhaitée.

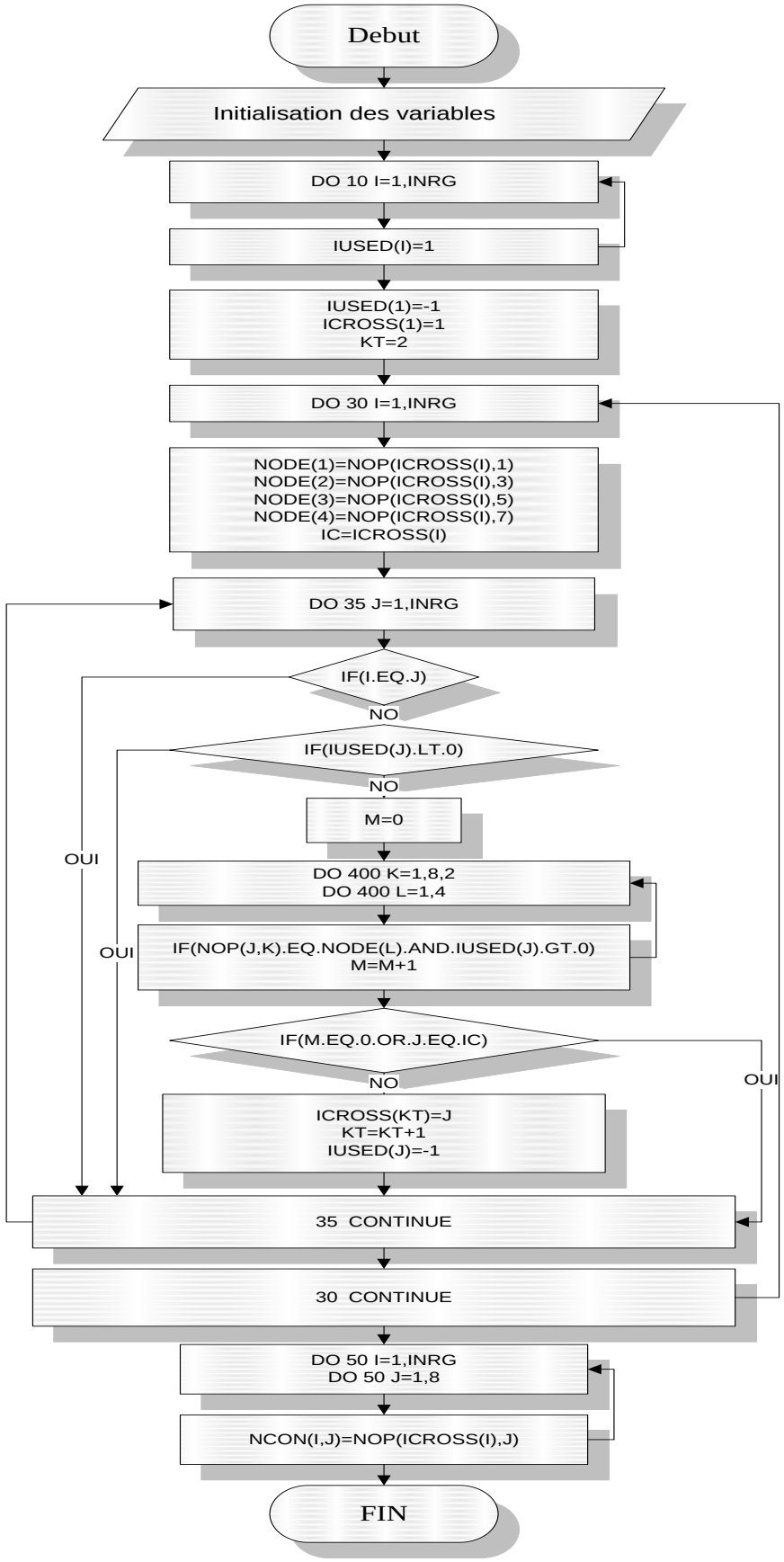


Figure (4.3) représente l'organigramme du sous-programme REORDR

4. 3.2.3 Le sous-programme MESH

Cette partie de la sortie de codage produit toutes les données d'entrée et génère tous les nombres de nœud, les coordonnées nodales, les épaisseurs et les températures, L'organigramme du sous-programme est illustré à la figure (4.4).

Le sous-programme **MESH** est appelé par le programme principal pour raffiner les super-éléments.

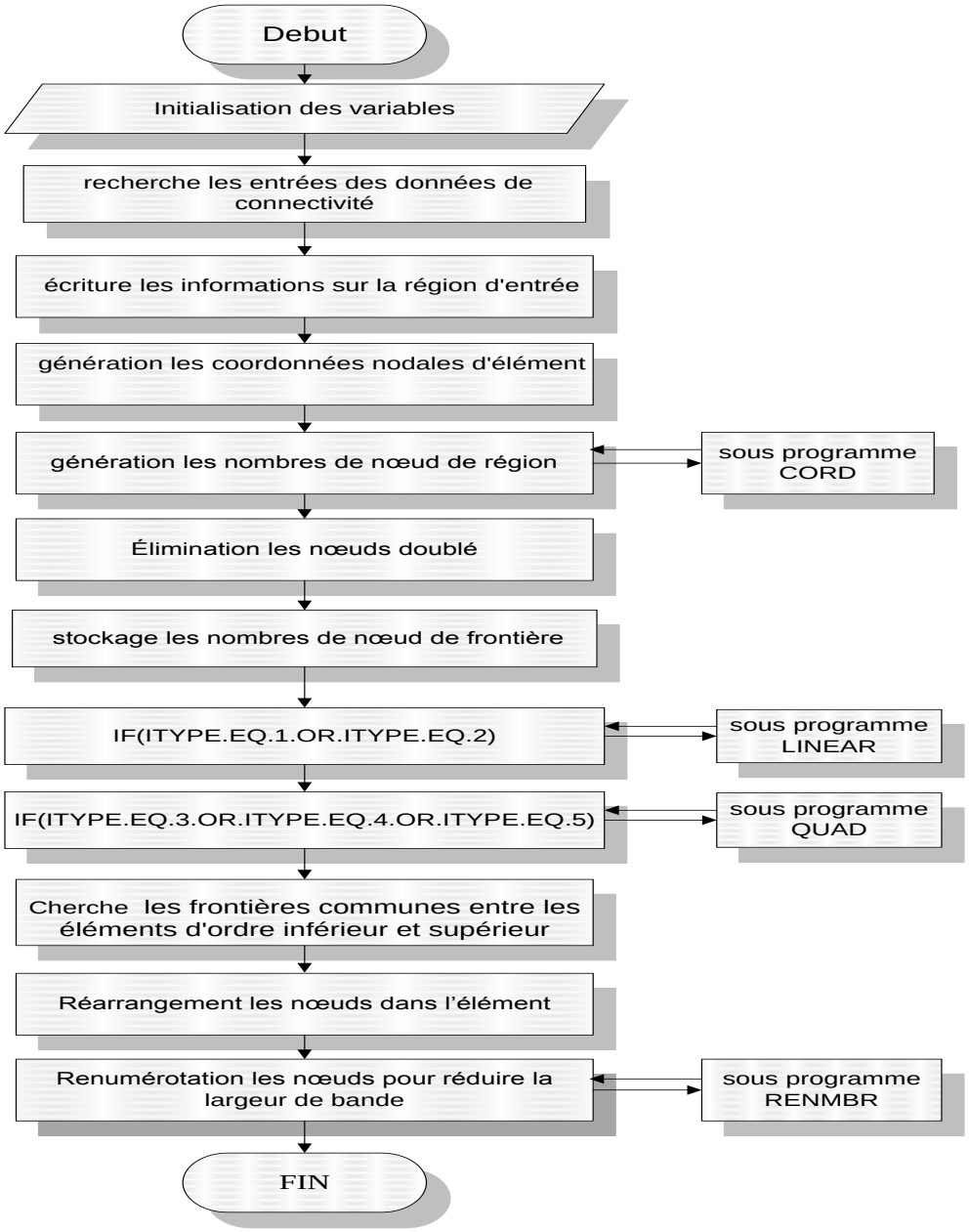


Figure (4.4) représente l'organigramme du sous-programme MESH

4.3.2.4 Le sous-programme LINEAR :

Le sous-programme **LINEAR** génère un groupe de connectivité des éléments et spécifie les codes des matériaux pour les triangles à 3 nœuds et quadrilatères à 4 nœuds, L'organigramme du sous-programme est illustré à la figure (4.5).

Les fonctions de forme dans le triangle

Les coordonnées naturelles (ξ,η) dans le triangle, alors les fonctions de forme peut être représenter simplement par :

$$N_1 = \xi, \qquad N_2 = \eta, \qquad N_3 = 1 - \xi - \eta \tag{4.6}$$

Notons que, $N_1 + N_1 + N_1 = 1$ (4.4)

Comme dans le cas 1-D, $N_1 = \begin{cases} 1 & \text{au noed i} \\ 0 & \text{à d'autre noeud} \end{cases}$ (4.8)

Les fonctions de forme dans le quadrilatère à 4 nœuds

Il y'a quatre nœuds aux sommets de la forme quadrilatéral. Dans le système de coordonnées naturelles (ξ,η), les quatre fonctions de la forme sont :

$$N_1 = \frac{1}{4}(1-\xi)(1-\eta) \quad N_2 = \frac{1}{4}(1+\xi)(1-\eta)$$
$$N_3 = \frac{1}{4}(1+\xi)(1+\eta) \quad N_4 = \frac{1}{4}(1-\xi)(1+\eta) \tag{4.9}$$

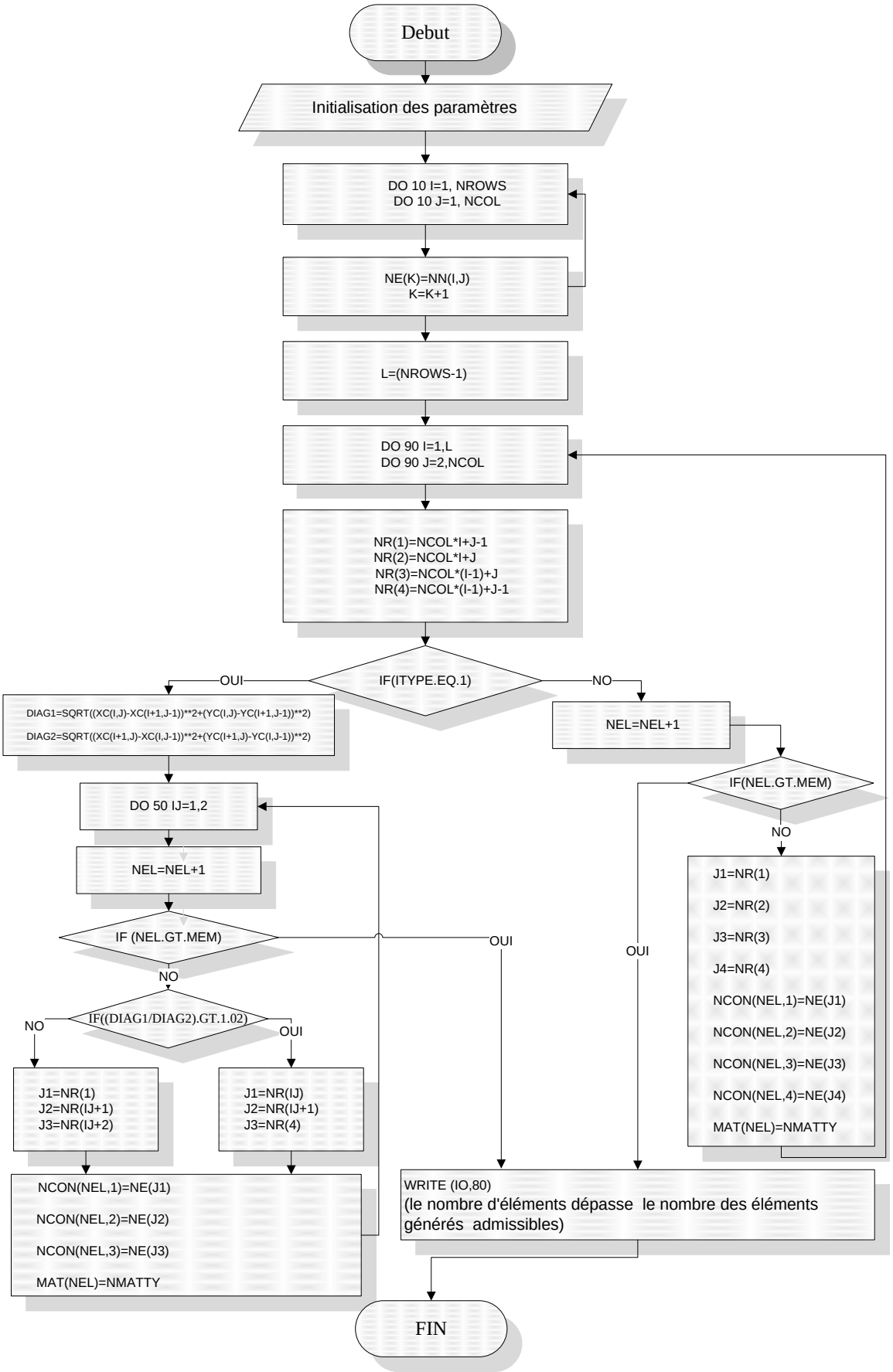


Figure (4.5) représente l'organigramme du sous-programme LINEAR

4.3.2.5 Le sous-programme QUAD

Le sous-programme **QUAD** établit les mêmes informations pour les éléments quadratiques (génère un groupe de connectivités des éléments et spécifie les codes des matériaux), les triangles à 6 nœuds de côté droites et courbes et le quadrilatère à 8 nœuds iso-paramétriques, l'organigramme du sous-programme est illustré à la figure (4.6).

Pour les éléments triangulaires, la connectivité est établie de manière à relier le long de la diagonale plus courte du quadrilatère.

Les sous-programmes LINEAR et QUAD sont appelés dans le sous-programme MESH pour générer des éléments linéaires et quadratiques, respectivement.

Les fonctions de forme dans le triangle à 6 nœuds

Cet élément est aussi appelé élément triangulaire quadratique.

Il y'a six nœuds dans cet élément. Chaque nœud à 2 degré de liberté

Dans le système de coordonnées naturelles, les fonctions de forme pour l'élément :

$$\begin{array}{lll} N_1 = \xi(2\xi - 1) & N_2 = \eta(2\eta - 1) & N_3 = \xi(2\xi - 1) \\ N_4 = 4\xi\eta & N_5 = 4\eta\xi & N_6 = 4\xi\xi \end{array} \qquad (4.10)$$

Les fonctions de forme d'élément quadrilatéral quadratique (Q8) :

C'est le plus largement utilisé des éléments pour les problèmes 2-D du à sa haute précision dans l'analyse.

Il y'a huit nœuds pour cet élément, quatre sommets et quatre au milieu.

Dans le système de coordonnées naturelles (ξ,η), les huit fonctions de forme sont,

$$\begin{array}{ll} N_1 = \frac{1}{4}(1-\xi)(1-\eta)(\xi+\eta+1) & N_3 = \frac{1}{4}(1+\xi)(1+\eta)(\xi+\eta-1) \\ N_2 = \frac{1}{4}(1+\xi)(1-\eta)(\eta-\xi+1) & N_4 = \frac{1}{4}(1-\xi)(1+\eta)(\xi-\eta+1) \\ N_5 = \frac{1}{2}(1-\eta)(1-\xi^2) & N_7 = \frac{1}{2}(1+\eta)(1-\xi^2) \\ N_6 = \frac{1}{2}(1+\xi)(1-\eta^2) & N_8 = \frac{1}{2}(1-\xi)(1-\eta^2) \end{array} \qquad (4.11)$$

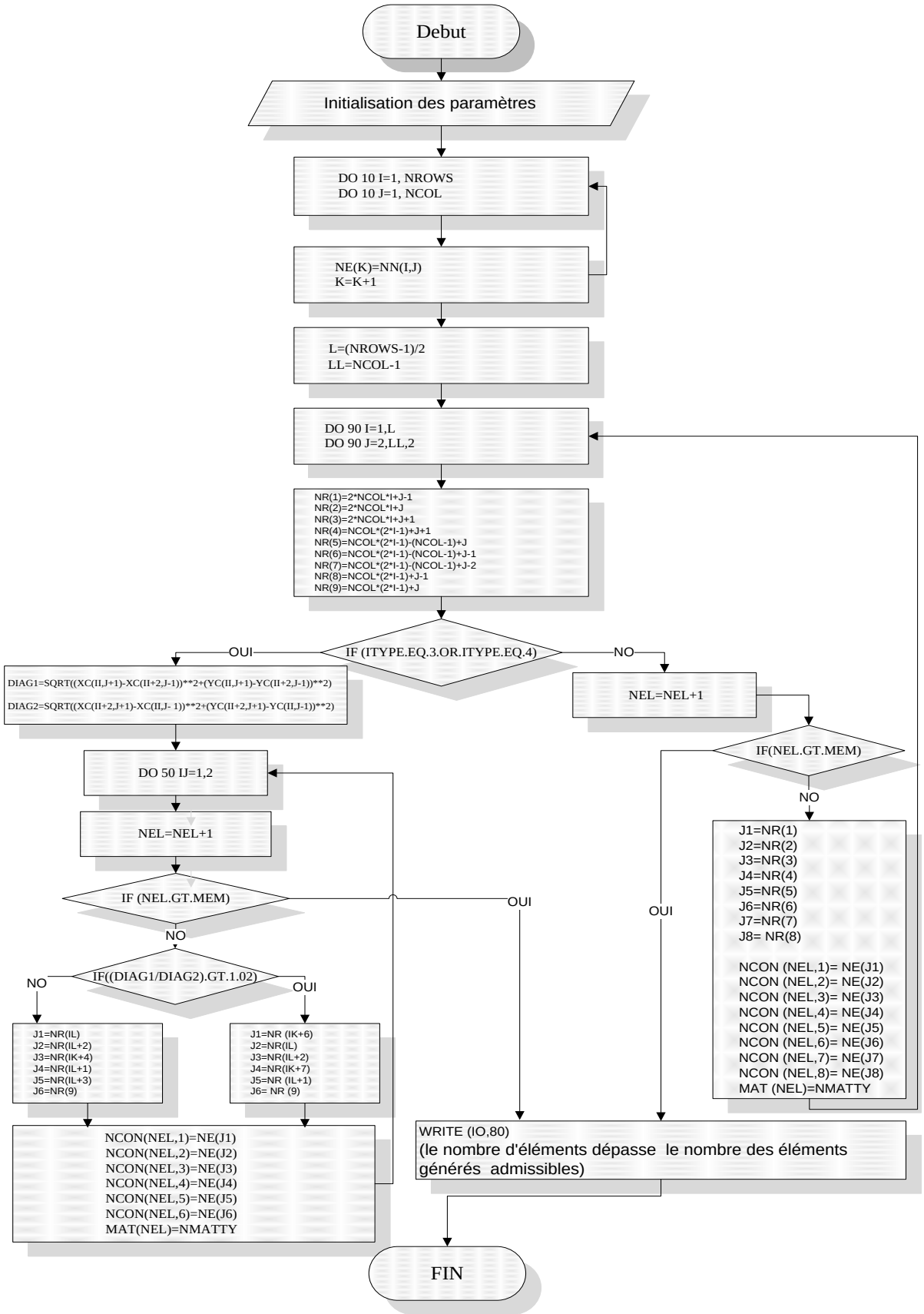


Figure (4.6) représente l'organigramme du sous-programme QUAD

4. 3.2.6 Le sous-programme CORD :

Est utilisé pour éviter la construction inexacte de maillage des éléments triangulaires le long de la direction diagonale. Les nœuds des éléments de différents types sont réarrangés et un maillage de différents d'ordre d'éléments est finalement construit dans le programme principal, L'organigramme du sous-programme est illustré à la figure (4.7).

Le sous-programme CORD est appelé par Le sous-programme MESH

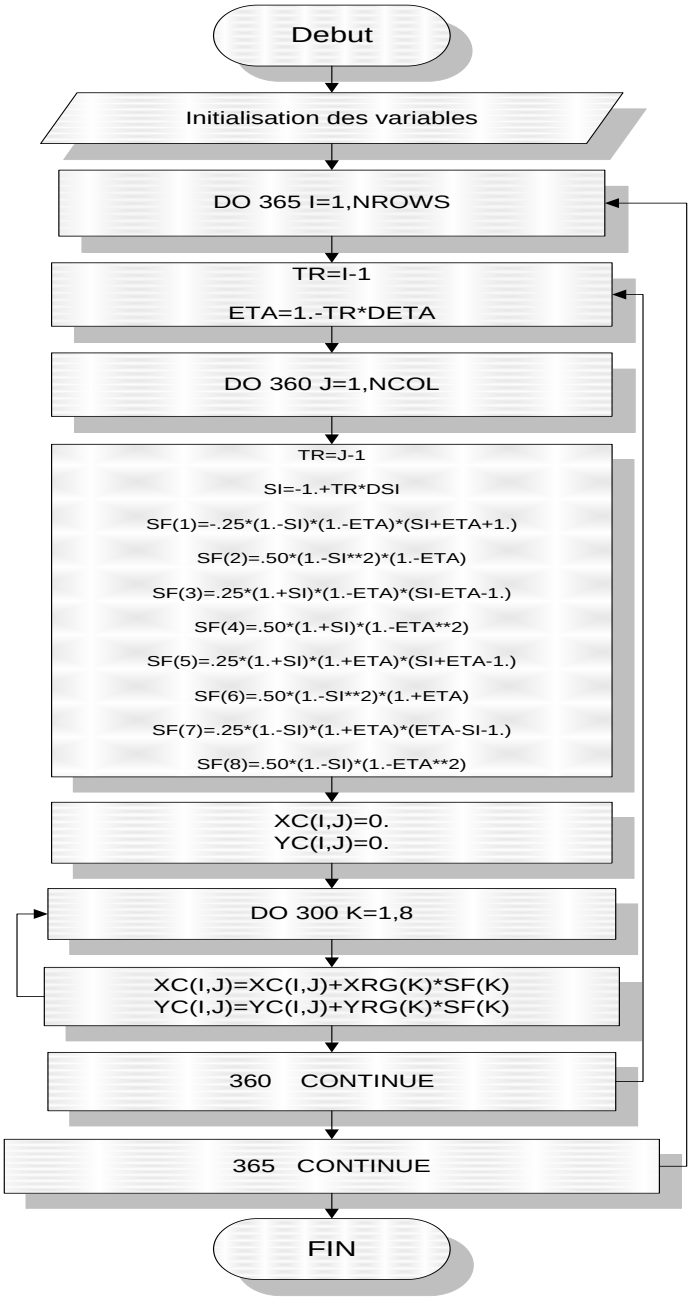


Figure (4.7) représente l'organigramme du sous-programme CORD

4.3.2.7 Sous-programme RENMBR, OPTNUM

Ces sous-programmes sont associés à une renumérotation nodale et peuvent être négligées si la renumérotation n'est pas nécessaire.

RENMBR détermine combien des nœuds (le nombre de nœud) sont connectés directement à un nœud donné et les numéros des nœuds qui sont connectés à un nœud donné. On suppose que n'importe quel nœud de donnée ne sera pas relié à plus de **MAXCON** (nombre des nœuds maximum peut être partagée pour chaque super-élément) avec les autres nœuds.

OPTNUM essai de "N" renumérotation où «N» est le nombre de nœuds dans le modèle. Certainement, l'utilisateur peut facilement remplacer son système clair de numérotation ou d'utiliser un autre algorithme.

Le sous-programme RENMBR doit être appelé dans MESH et le sous-programme OPTNUM alors sera appelé par RENMBR.

4.3.2.8 Sous-programme SORTIE

L'information de sous-programme SORTIE se compose du nombre total des nœuds générés, le nombre total des éléments générés, le nombre de nœuds liés à un élément donné, les coordonnées nodales, la connectivité d'élément et les code de matériau, les températures nodales et les épaisseurs. Et la différence maximale de nœud.

4.3.3. Notations du programme (MESH_2D):

INRG ou **NRG** - (nombre de super-éléments)
INBP - (nombre de nœuds d'entrée)
IPOL - (Nombre de systèmes de coordonnées polaires auxiliaires)
ITMP - (Option de la température d'entrée)
ITHK - (Option de l'épaisseur d'entrée)
IRENUM - (Option Renuméroter)

(A) 0 - (ne renuméroter pas)

(B) 1 - (Renuméroter)

IPOLAR - (option des systèmes de coordonnées)

(A) 0 - (Systèmes de coordonnées cartésiennes)

(B) 1 - (les numéros supérieurs à 0 - différents systèmes de coordonnées polaire auxiliaires)

NOP - (Tableau représentant le nombre de nœud de toutes les entrées de super-éléments)

ITYPE - (tableau représentant le type d'éléments de sortie) :

(A) 1 ------- (éléments triangulaire à 3 nœud)

(B) 2 ------- (éléments linéaire à 4 nœud)

(C) 3 ------- (éléments isoparamétriques à 6 nœud)

(D) 4 ------- (éléments triangulaire à 6 nœud)

(E) 5 ------- (éléments isoparamétriques à 8 nœud)

XP (K, 1) - (tableau représentant les coordonnée x d'entrée)

YP (K, 1) - (tableau représentant les coordonnée y d'entrée)

XP (K, 2) - (Tableau représentant la température d'entrée)

YP (K, 2) - (tableau représentant l'épaisseur d'entrée, ou d'autres propriétés)

NROW1 - (Tableau représentant la division dans la direction (**η**) **ETA** pour chaque super-élément)

NCOL1 - (Tableau représentant la division dans la direction (**ξ**) **XI** pour chaque super-élément)

IND11(NRG) ou **IND1** - (indicateur des nœuds de frontière de chaque d'entrée de super-éléments)

NMAT - (Tableau représentant le code matériau de chaque entrée de super-éléments)

NB - (Nombre de nœuds générés)

NUMEL ou **NEL** - (Nombre d'éléments générés)

NCN - (tableau représentant le nombre de nœuds liés à tout élément généré)

NNRB - (tableau généré le stockage de tous les nœuds limite de super-éléments)

IVOE - (option d'ordre de maillage d'élément)

(A) 0 ------- (maillage d'ordre uniforme)

NJT - (nombre maximum de nœuds générée)

MEM - (nombre maximum d'éléments générés)

NM -nombre maximum de lignes ou des colonnes générées dans un super-élément

MAXCON - (nombre des nœuds maximum peut être partagée pour chaque super-élément)

IDIFF - (différence de nœud maximum dans tous les éléments générés)

IN - (nombre d'équipement d'entrée (écran, fichier, etc.))

IO - (nombre d'équipement de sortie (écran, fichier, etc.))

4.4 Description du programme MAT-fem:

L'organigramme du sous programme du fichier de données est donné par la figure (4.8), est écrit en "MATLAB".

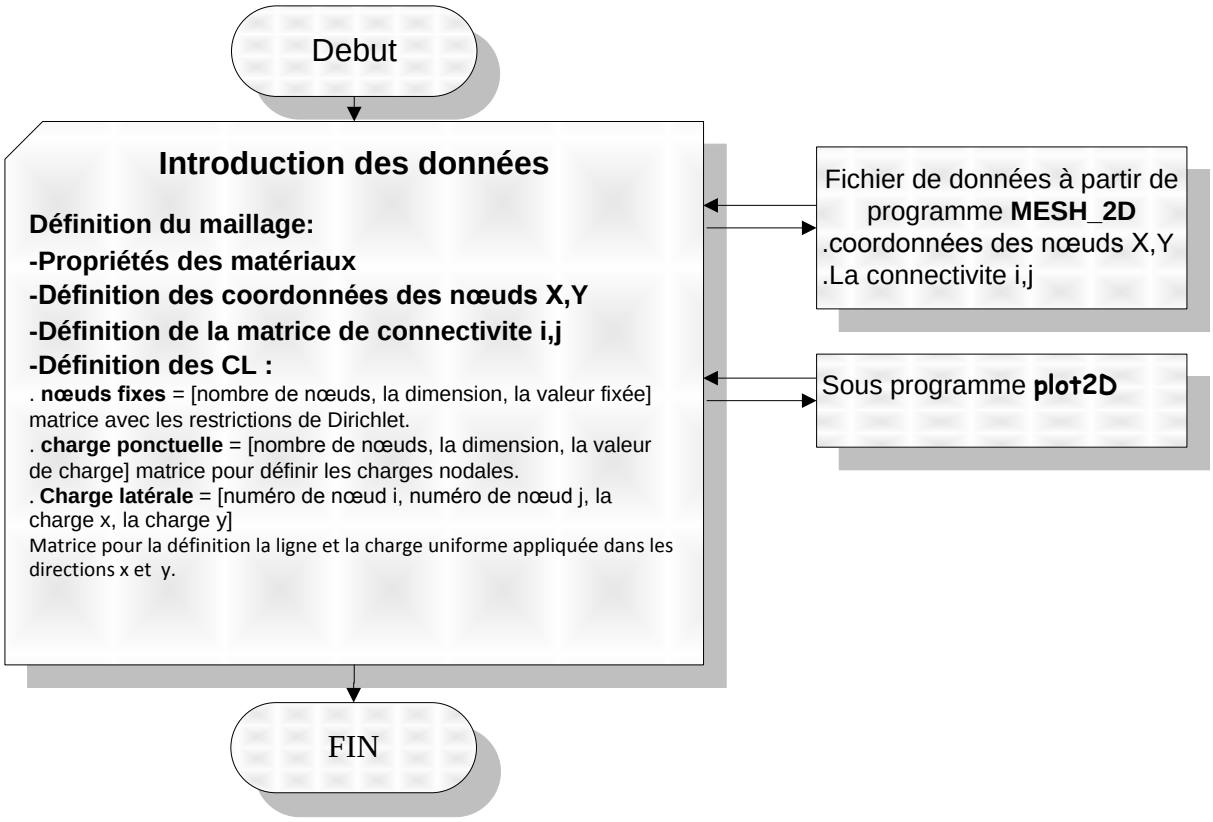


Figure (4.8) représente sous programme du fichier de données.

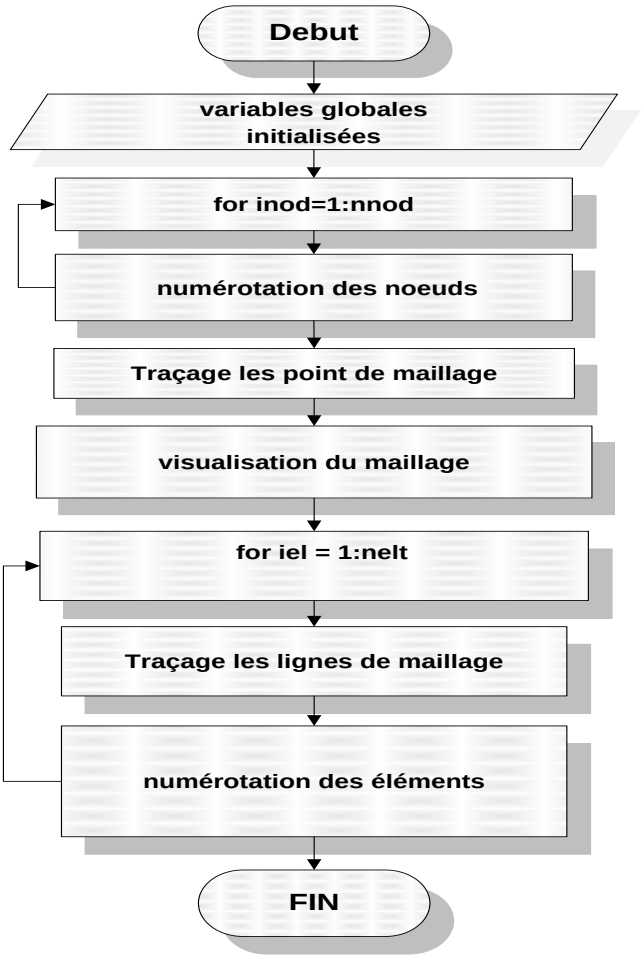


Figure (4.9) représente sous programme plot2D

La visualisation regroupe l'ensemble des techniques qui permettent de représenter les quantités physiques sous une forme graphique pour faciliter la compréhension des résultats de maillage et pour faciliter à introduire les conditions aux limites. Le sous programme **plot2D** tracé du maillage d'une structure plane ou bien des géométries complexe avec les nombre des nœuds et des éléments ,le sous programme est écrit par le langage de MATLAB en peut visualiser la numérotation des nouds et la numérotations des éléments , L'organigramme du sous-programme est illustré à la figure (4.9).

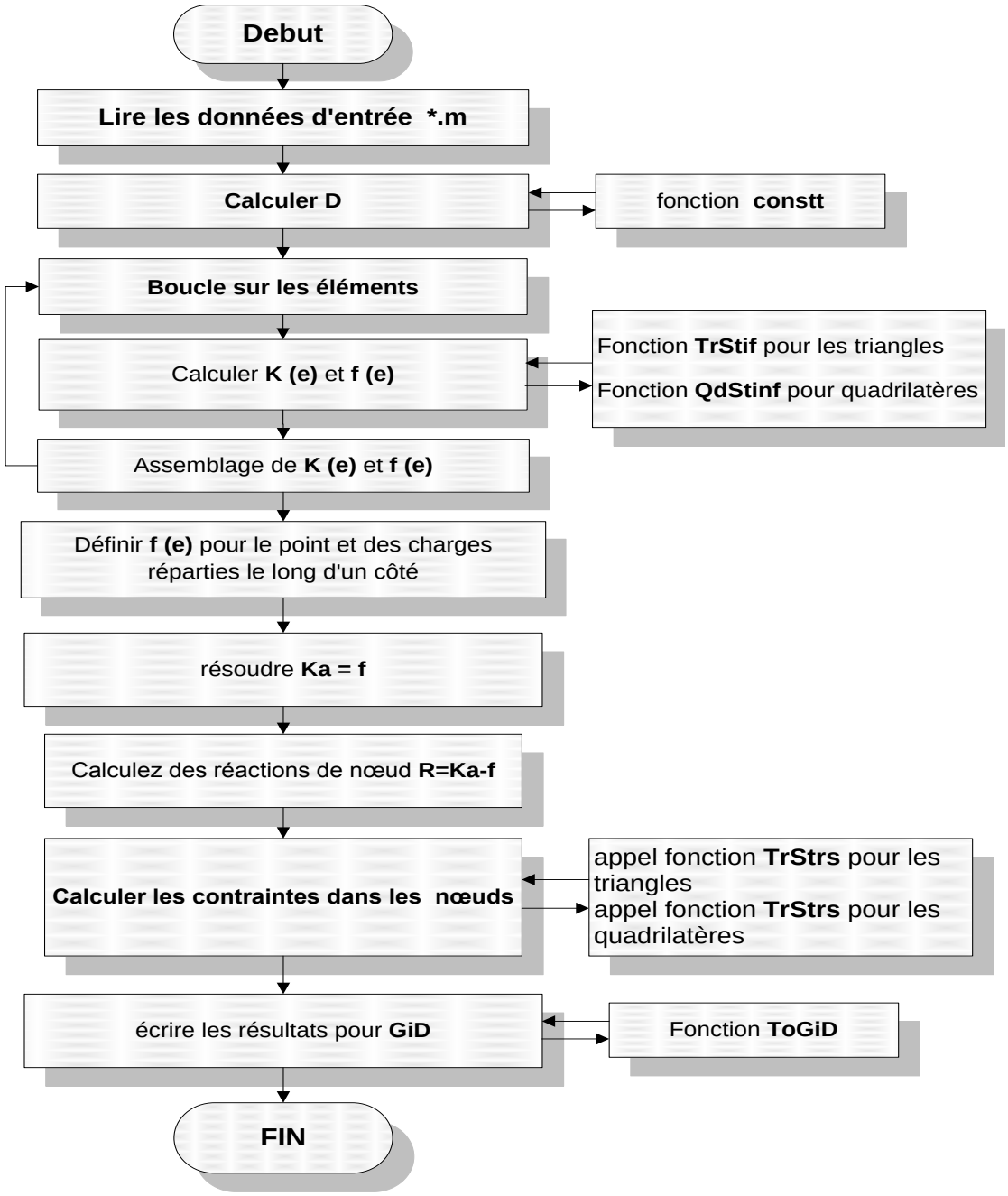


Figure (4.10) représente L'organigramme du programme MAT-fem

Comme toutes les autres méthodes numériques, la méthode des éléments finis est liée à des algorithmes de programmation qu'ils le définissent. En effet, son avantage est dans l'itération des calculs sur une grande quantité d'éléments. Historiquement, le premier langage de programmation utilisé pour travailler avec la méthode des éléments finis a été le FORTRAN; depuis lors, de nombreuses sous programmes, des algorithmes et des programmes associés à la méthode ont été programmés dans cette langue (FORTRAN).

Avec le développement de l'informatique, des langues nouvelles sont apparues, chacun avec des capacités et des outils spécifiques pour des domaines d'application divers, tout cela avec l'idée de simplifier la codification des algorithmes et optimiser les ressources informatiques.

MAT-fem est un programme d'exécution de haut en bas. L'organigramme du programme est illustré à la Figure (4.10). Le module d'entrée de données est mis en œuvre dans le même dossier ont été données est définie, tel que décrit dans la section suivante.

Pour un système **discret** (Élasticité, Thermique,...), les équations de comportement peuvent en général s'écrire sous la forme matricielle suivante :

[K] {U} = {F}

[K] - matrice caractérisant le système

{U} - variables inconnues du problème

DDL – Degré De Liberté

DOF – Degree Of Freedom

{F} - sollicitations connues (second membre)

Problème	Propriété [K]	Comportement {U}	Action {F}
Élasticité	rigidité	déplacement	force
Thermique	conductivité	température	flux de chaleur

4.4.1 Début :

MAT-fem commence à mettre toutes les variables égales à zéro avec la commande **clear**. Ensuite, il demande à l'utilisateur le nom du fichier de données d'entrée qu'il va utiliser (l'extension .m n'est pas inclus dans le nom du fichier) , L'organigramme d'un sous-programme du fichier de données est illustré à la figure (4.8).

Les premières lignes du code correspondant au coffre des variables ainsi que l'horloge mis en place, qui stocke le temps total d'exécution dans **ttim**.

La lecture des données, est une tâche d'allocation de variable directe dans le programme (Fichier de données à partir de programme MESH_2D .les coordonnées des nœuds X,Y et La connectivite i,j) .

D'après les données des matrices, il est possible d'extraire les dimensions de base du problème, comme le nombre de points nodaux, **npnod**, ce qui correspond au nombre de lignes dans les coordonnées de la matrice. Le nombre total de degrés de liberté du problème, **nndof**, sera le double du nombre de nœuds (**2*npnod**). **nelem** est le nombre d'éléments et est égal au nombre de lignes dans les éléments de la matrice, alors que le nombre de nœuds pour chaque élément **nnode** est le nombre de colonnes dans les éléments. De cette façon, les éléments triangulaires sont identifiés si le nombre de colonnes dans les éléments est de trois, tandis que le nombre de colonnes sera de quatre pour les éléments quadrilatère.

Le nombre total d'équations par élément, **neleq**, est le nombre de nœuds d'élément **nnode** multiplié par le nombre de degrés de liberté pour chaque nœud, soit **deux** pour des problèmes 2D. Il est à noter que ces variables sont définies dans la structure de données, ce qui simplifie l'interprétation du code.

Dans tout le programme, le sous-programme de chronométrage est utilisé pour calculer le temps de course entre deux déclarations dans le code. De cette façon l'utilisateur peut vérifier les sections du programme qui nécessitent des efforts de calcul. À l'intérieur du chronométrage des commandes du MATLAB **tic** et **toc** sont utilisé.

4.4.2 Matrice de rigidité et vecteur de force nodal équivalent

4.4.2.1Généralités

MAT-fem utilise des algorithmes très puissants sans perdre sa simplicité. L'objectif principal du programme est de démontrer la mise en œuvre du FEM, certaines simplifications sont faites comme l'utilisation d'un seul matériau pour l'ensemble du domaine. Par conséquent, la matrice constitutive ne varie pas entre les éléments adjacents et il est évalué avant de lancer le calcul de la matrice de rigidité d'élément.

Le sous-programme **constt** implique l'utilisation du module de Young, coefficient de Poisson et un signe qui permet de distinguer entre une contrainte plane et un problème de déformation plane pour former la matrice constitutive qui est stockée dans **dmat**. Le sous-

programme **constt** montre la forme explicite de **dmat** pour un matériau élastique linéaire isotrope et ce dans les problèmes d'élasticité linéaire.

Par contre, dans les problèmes thermique Le sous-programme **constt** implique l'utilisation du la conductivité thermique.

MAT-fem recalcule les valeurs de chaque variable au lieu de les stocker. Le processus de recalcule est exécuté de façon rapide et ne réduit pas significativement l'efficacité du programme. Cela laisse plus de mémoire pour résoudre de plus grands problèmes.

Le programme **MAT-fem** montre que la boucle d'élément dans laquelle la matrice de rigidité et le vecteur nodal équivalent pour les poids propre sont calculées et assemblées pour chaque élément. La boucle commence à récupérer les propriétés géométriques pour chaque élément.

Le vecteur **lnods** stocke les connectivités nodales pour l'élément et la matrice **coord** stocke les coordonnées pour ces nœuds.

Les sous-programmes **TrStif** et **QdStif** sont appelés pour les éléments triangulaires et quadrangulaires, respectivement. Le même sous-programme évalue la matrice de rigidité de l'élément et le vecteur de force nodal équivalent pour l'élément. L'utilisation de la même intégration quadrature permet cette simplification.

Le calcul de la matrice de rigidité d'élément pour chacun des deux éléments considérés est détaillé dans la section suivante.

Avant l'assemblage, le **vecteur d'équations eqnum** est défini. Il contient le nombre global d'équations dans la matrice de rigidité d'élément.

Le processus d'assemblage équations est mis en œuvre au moyen de deux boucles de 1 à **neleq** (nombre d'équations pour chaque élément). Dans la première boucle le vecteur de force nodal équivalent est assemblé et dans le deuxième, la matrice de rigidité de l'élément est assemblé terme à terme. Ce système évite de stocker temporairement les matrices d'élément.

4.4.2.2 Calcul et assemblage de K (e) et f (e) (poids propre) pour les triangles à 3 nœuds et des quadrilatères à 4 nœuds :

La matrice de rigidité pour le triangle à 3 nœud est calculée sous forme explicite tandis que pour le quadrilatère à 4 nœud est calculée par intégration numérique. Les deux sous programmes nécessitent exactement les mêmes données d'entrée et retour aux mêmes variables:

ElemMat pour la matrice de rigidité et **ElemFor** pour le vecteur de force nodale équivalente.
Le sous-programme **TrStif** (appelé comme fonction) est défini pour les éléments triangulaires. Il est à noter que les dérivés cartésiens des fonctions de forme linéaire sont calculés directement.

La matrice de rigidité de l'élément est calculée par l'expression classique $\mathbf{B^T DB\ dA}$ et stockées dans la matrice **M**. L'un des avantages fondamentaux de MATLAB est d'observer que la multiplication des matrices est exécutée au moyen d'une instruction unique, éliminant le besoin d'écrire des boucles gênantes.

Le programme MAT-fem montre aussi l'évaluation de la matrice de rigidité de l'élément quadrilatère qui est exécutée en utilisant l'intégration numérique. Au départ, les fonctions de forme des éléments et leurs dérivés naturels sont définis (**fform** et **Deriv**).

Cette définition est faite au moyen d'une fonction intrinsèque. Cette facilité de MATLAB évite l'utilisation de sous-programmes supplémentaires.

À ce niveau aussi les valeurs de coordonnée du point de Gauss **(pospg**) et leurs poids (**pespg**) correspondant à la règle d'intégration (2*2) sont définies. On montre aussi la boucle sur les points de Gauss pour calculer la matrice de rigidité et le vecteur de charge nodal équivalent pour l'élément**.**

Une fois les variables sont initialisées, deux boucles définissent le processus d'intégration de Gauss. Le **vecteur lcffm** contient les valeurs des fonctions de forme évaluées au point d'intégration (i, j) et la **matrice lcder** stocke les valeurs pour les dérivées naturelles des fonctions de forme à chaque point d'intégration.

La matrice jacobienne **(xcjacm**) est évaluée par la multiplication de la matrice **lcder** par les coordonnées des nœuds d'élément. Les valeurs des dérivées cartésiennes des fonctions de forme **(ctder)** à chaque point d'intégration sont obtenues en multipliant l'inverse de la matrice **jacobienne** par les dérivées naturelles des fonctions de forme calculées au point d'intégration.

Le différentiel de **surface (darea**) est calculé comme le déterminant de la matrice jacobienne multipliée par les deux fonctions de pondération au point d'intégration et l'épaisseur d'élément. La matrice de déformation est obtenue en plaçant les dérivés cartésiens des fonctions de forme dans le tableau matriciel **bmat**.

La matrice de rigidité d'élément est obtenue en intégrant l'expression standard $\mathbf{B^TDB}$.

L'intégration numérique exige l'évaluation de la somme du produit de $\mathbf{B^TDB}$ et le calcul de **darea** à tous les points de Gauss. Cette somme est stockée dans la matrice **M** .

Le vecteur de force nodal équivalent exige l'intégration sur la surface d'élément par le produit des fonctions de forme et le poids spécifique pour l'élément. Le vecteur est stocké dans la variable **Fy**.

Finalement les composants de force nodaux équivalents sont placés dans le vecteur **F**. Les sous programmes **TrStif** et **QdStif** montrent clairement les deux choix pour calculer la matrice de rigidité d'élément l'utilisation de la forme explicite (pour le triangle à 3 nœuds) ou au moyen de l'intégration numérique (pour le quadrilatère à 4 nœuds).

4.4.3 Charges externes :

En plus de la charge de poids propre on considère des charges uniformément distribuées agissant sur les côtés d'élément et les charges de point nodales. À la fois les deux éléments considérés à des fonctions de forme linéaires, le calcul de la contribution nodale pour des charges latérales uniformément distribuées est exactement le même pour les deux cas. L'évaluation est faite dans le sous programme principal du programme **MAT-fem** après l'assemblée de la matrice de rigidité.

4.4.4 Solutions du système d'équations :

La stratégie utilisée dans **MAT-fem** consiste essentiellement à résoudre le système

d'équations global sans tenir compte de ces degrés de liberté dont les valeurs sont connues (c'est-à-dire imposées).

Le vecteur **FreeNodes** contient la liste des équations à résoudre. Le vecteur **FreeNodes** est utilisé comme un indice de DOF et permet d'écrire la solution du système d'équations d'une manière simple. MATLAB prend soin de choisir l'algorithme le plus approprié pour résoudre le système. L'étape de solution est totalement transparente pour l'utilisateur. Les sous programmes mises en œuvre dans le noyau MATLAB contribuent de nos jours dans l'optimisation de la vitesse de mémoire avec les meilleurs algorithmes existants.

4.4.5 Calcul des réactions nodales :

La solution du système d'équations est stockée dans le vecteur **U** contenant les déplacements nodaux dans le domaine élasticité linéaire mais dans le domaine thermique **U** représente la température. Les réactions nodales sont calculées au moyen de l'expression: reaction=StifMat*u-force.

Évidemment, la valeur des réactions au niveau des nœuds imposés n'est pas nulle.

Pour éviter des calculs inutiles, on utilise le vecteur fixé qui contient la liste des équations associées aux degrés de liberté imposés**.**

Remarque :

Dans le domaine thermique on utilise le terme « flux réactif » au lieu de « réaction »

4.4.6 Calcul des contraintes au niveau des nœuds :

Le calcul de contrainte dans les éléments nécessite l'utilisation d’un sous-programme, non seulement pour le calcul stress lui-même, mais aussi projeter les contraintes des points d'intégration aux nœuds.

Le sous-programme **stress** contrôle le flux du programme vers les sous programmes d’élément. Pour l'élément à 3 nœuds triangulaires, les contraintes sont constantes et l’extrapolation nodale est triviale.

Ce n'est pas le cas pour l'élément quadrilatère à 4 nœuds où les contraintes sont une variation bilinéaire et la contrainte d'extrapolation est effectuée en utilisant les fonctions de forme de l'élément.

Le programme MAT-fem montre les étapes générales pour le calcul des contraintes au niveau des nœuds d'un maillage de triangles à 3 nœuds et des quadrilatères à 4 nœuds en moyenne nodale.

Pour les éléments triangulaires le sous programme construit le vecteur **ElemStr** qui contient les contraintes nodales pour l'élément. Ces valeurs sont accumulées dans **nodstr** dont la dernière colonne est le nombre d'éléments qui partagent le nœud. Ceci est nécessaire afin d'évaluer la moyenne nodale des contraintes dans une étape finale.

Pour les éléments quadrilatères **ElemStr** est une matrice qui contient les contraintes à chacun des nœuds d'élément. Comme dans le cas précédent. La dernière colonne de l'ElemStr contient le nombre d'éléments qui partagent le nœud. Une fois les contraintes à tous les nœuds d'élément ont été calculées et accumulées dans **nodstr**, une moyenne nodale est effectuée pour calculer un champ de contraintes lissée au niveau des nœuds.

4.4.7 Etape de post-traitement

Une fois les déplacements nodaux, les réactions et les contraintes ont été calculées, les valeurs sont transférées vers les fichiers de post-traitement, d'où le logiciel **GiD** pourra les afficher sous forme graphique. Elle est réalisée dans le sous programme **ToGiD**. Au cours de l'exécution du programme le temps total utilisé par le programme apparaît dans la console MATLAB ainsi que le temps consommé dans chaque sous-programme. La plus forte consommation de temps dans la majorité des problèmes résolus avec MAT-fem est investie dans le calcul et l'assemblage de la matrice de rigidité globale, alors que la solution du système d'équations représente un faible pourcentage du temps consommé. Ce n'est pas généralement le cas pour la solution des grands problèmes pour les structures de taille réelle. Une fois l'exécution du programme est terminée, les variables sont toujours enregistrées dans MATLAB pour expérimenter avec l'ensemble des fonctions internes disponibles.

4.4.8 Description Gid:

Gid est un logiciel conçu dans le but de couvrir toutes les applications liées au domaine de la simulation numérique. Il est utilisé pour les phases de pré et de post traitement. Le développement de **GID** a été axé sur les besoins de l'utilisateur en termes d'efficacité, rapidité et simplicité.

Gid s'adapte facilement aux codes de simulation numérique. En effet, **Gid** offre un choix étendu sur les formats d'entrée et de sortie. Les principaux avantages de **Gid** sont les suivants :

- Il réduit le temps et les coûts associés à l'analyse numérique grâce à sa performance d'acquisition des données d'entrée et sa haute qualité de post-traitement.

- Il peut être utilisé directement à partir d'un logiciel de **CAO**. Il n'est donc pas nécessaire de reconstruire la géométrie ce qui permet un gain de temps précieux.

- Sa simplicité d'utilisation, grâce à une interface intuitive, permet de se former rapidement.

On distingue 3 zones :

- La zone 1 est le menu général permetant d'importer et d'exporter les différents fichiers. Elle permet d'imprimer les résultats et de prendre des captures d'écran.

- La zone 2 regroupe les outils nécessaires au traitement des données. Les outils changent en fonction qu'il s'agisse du pré ou du post traitement. Il s'agit ici des outils du post traitement.

- La zone 3 permet la visualisation du travail en cours.

4.4.9 Limites de calcul actuelles du générateur de maillage.

Le mailleur est capable de traiter, à l'heure actuelle, des études comportant au maximum :

NJT - (nombre maximum de nœuds générés)= 9000
MEM - (nombre maximum d'éléments générés)= 6000

Mais en peut augmenter les 2 nombre comme notre besoin de calcule et la limite de programme MAT-fem et dépendre de la limite de notre mailleur.

5. 4- Propriétés essentielles d'un mailleur :

En fin, et avant de clore ce chapitre, on présente ici un bref aperçu sur les principales propriétés d'un mailleur.

En effet, un bon mailleur pour éléments finis doit tenir compte de ces considérations de base :

1. La qualité de la solution obtenue lors de l'analyse par éléments finis dépend de la finesse du maillage produit.

Cela signifie que plus le maillage est fin (i.e. la concentration en éléments de maillage est élevée), plus précise sera la solution obtenue, ceci étant particulièrement vrai dans les zones où les gradients de solution sont importants.

2. Les coûts d'analyse augmentent généralement beaucoup plus vite que la finesse de maillage.

En d'autres termes, les coûts d'analyse (cette notion de *coût* recouvre des paramètres tels que le temps de calcul sur ordinateur, la place mémoire nécessaire, les temps de pré- et post- traitement, etc.) peuvent rapidement devenir prohibitifs si l'on tente d'appliquer uniformément un maillage très fin .

Une conséquence immédiate de ces considérations est que pour obtenir des solutions de bonne qualité, il faut pouvoir préciser les niveaux de maillage de façon non uniforme. **On doit pouvoir imposer un maillage fin dans les zones susceptibles d'être le siège de forts gradients de la solution, et un maillage plus grossier dans les zones où les solutions n'évoluent que peu**. Les éléments de maillage générés, en principe, des entités géométriques simples (triangles et quadrilatères en 2D et tétraèdre, pyramide, prisme et hexaèdre en 3D), doivent donc avoir les tailles requises par les paramètres de maillage dans les zones où ils apparaissent.

3. Le maillage généré doit être le plus fidèle possible à la géométrie du problème. En particulier, les différents contours extérieurs et intérieurs qui apparaissent une fois le maillage

généré doivent être aussi proches que possible des frontières originales.

Certains mailleurs vont générer des maillages tels que les sommets de certains éléments vont se trouver le long d'un côté d'un élément voisin. Ces mailleurs vont devoir mettre en œuvre des méthodes destinées à résoudre les problèmes de continuité que de telles configurations entraîneraient.

4. Il convient également de contrôler la forme des éléments finis générés. En effet, les calculs effectués au moment de l'analyse par éléments finis sont faits sur des éléments de référence (triangle ou carré) qui sont ensuite déformés pour coller aux différents éléments de maillage analogues. Des éléments trop déformés par rapport à ces éléments de référence amèneraient des problèmes de types numériques (singularité, imprécision, ...).

5. Enfin, si le but premier du mailleur doit être de soulager l'analyse dans sa tâche de génération d'un maillage correct, il faudrait également avoir comme objectif à plus long terme de le faire travailler (i.e. essentiellement échanger des données) avec d'autres programmes (modélisation du ou des objets, saisie des paramètres physiques tels que les matériaux utilisés, définition des sollicitations appliquées, analyse et calculs, etc.) ; il convient donc de soigner l'aspect structure de données.

5. Applications :

Après avoir présenté, le programme MESH_2D (pour la génération de maillage en deux dimensions), nous nous proposons maintenant de l'appliquer à quelques problèmes rencontrés en génie mécanique.

Pour faire l'analyse d'une pièce mécanique, on doit calculer les contraintes et les déformations dans les problèmes de contraintes planes ou les températures et les flux de chaleur dans les problèmes thermique.

Dans ce chapitre, nous allons présenter des exemples d'application d'analyse par la méthode des éléments finis.

Pour ce faire, on a suivi la méthodologie suivante :

- ✓ Premièrement, on cherche à montrer la performance du code développé en utilisant quatre types d'éléments différents (soient les éléments linéaire triangulaires à 3 nœuds ou quadrangulaires à 4 nœuds ou bien les éléments quadratiques triangulaires à 6 nœuds ou quadrangulaires à 8 nœuds).
- ✓ Deuxièmement, on présente l'effet du type d'élément.
- ✓ Troisièmement, on expose la génération de maillage de quelques géométries complexes.
- ✓ Et à la fin, on montre l'adaptation de notre mailleur à un solveur **MAT-fem** pour traiter les problèmes de contrainte plane et les problèmes thermiques.

5.1 Mise en marche du code et comparaison avec les résultats des travaux [3,15] :

Après une mise en marche de notre code développé, on a par reproduire les résultats de [3,15] comme montrent les figures (5.1) et (5.2) pour les mêmes données (figure5.3).

```
VARIABLES = "X", "Y"
ZONE N=    19, E=    10 , F=FEPOINT , ET=QUADR
    -4.000000      2.000000
    -2.000000      2.000000
    -4.000000   0.000000E+00
    -2.000000   0.000000E+00
 -7.071068E-01   7.071068E-01
    -1.000000   0.000000E+00
 0.000000E+00      2.000000
 0.000000E+00      1.000000
    -4.000000      4.000000
    -2.000000      4.000000
 0.000000E+00      4.000000
     2.000000      4.000000
     2.000000      2.000000
 7.071068E-01   7.071068E-01
     4.000000      4.000000
     4.000000      2.000000
     2.000000   0.000000E+00
     4.000000   0.000000E+00
     1.000000   0.000000E+00
      3   4   2   1
      4   6   5   2
      5   8   7   2
      1   2  10   9
      2   7  11  10
      7  13  12  11
      8  14  13   7
     13  16  15  12
     17  18  16  13
     19  17  13  14
```

Figure (5.1) fichier de résultats de réf [3]

```
VARIABLES = "X", "Y"
ZONE N=    19, E=    10 , F=FEPOINT , ET=QUADRILATERAL
    -4.000000      2.000000
    -4.000000   0.000000E+00
    -2.000000   0.000000E+00
    -2.000000      2.000000
    -2.000000      4.000000
    -4.000000      4.000000
  0.000000E+00      4.000000
    -1.000000   0.000000E+00
 -7.071068E-01   7.071068E-01
  0.000000E+00      1.000000
  0.000000E+00      2.000000
     2.000000      2.000000
     2.000000      4.000000
  7.071068E-01   7.071068E-01
     4.000000      2.000000
     4.000000      4.000000
     2.000000   0.000000E+00
     4.000000   0.000000E+00
     1.000000   0.000000E+00
      2   3   4   1
      3   8   9   4
      9  10  11   4
      1   4   5   6
      4  11   7   5
     11  12  13   7
     10  14  12  11
     12  15  16  13
     17  18  15  12
     19  17  12  14
```

Figure (5.2) fichier de résultats obtenus par notre code mesh_2D

```
10 32 1 0 0 0 1
1 0. 0.
0 1 -4. 0.
0 2 -4. 2.
0 3 -2. 0.
0 4 -2. 1.
0 5 -2. 2.
0 6 -1.5 0.
1 7 1.9 135.
0 8 -1. 0.
1 9 1. 157.5
1 10 1. 135.
0 11 -1. 2.
1 12 1. 112.5
0 13 0. 1.
0 14 0. 1.5
0 15 0. 2.
0 16 -4. 4.
0 17 -2. 4.
0 18 0. 4.
1 19 1. 67.5
0 20 1. 2.
1 21 1. 45.
1 22 1.9 45.
0 23 2. 2.
0 24 2. 4.
1 25 1. 22.5
0 26 2. 1.
0 27 4. 4.
0 28 4. 2.
0 29 1. 0.
0 30 1.5 0.
0 31 2. 0.
0 32 4. 0.
1 2 2 2 1 3 5 2 1 5
2 2 2 1 3 6 8 9 10 7 5 4 2 5
3 2 2 1 10 12 13 14 15 11 5 7 3 5
4 2 2 2 2 5 17 16 4 5
5 2 2 2 5 15 18 17 5 5
6 2 2 2 15 23 24 18 6 5
7 2 2 2 23 28 27 24 7 5
8 2 2 2 31 32 28 23 8 5
9 2 2 1 29 30 31 26 23 22 21 25 9 5
10 2 2 1 13 19 21 22 23 20 15 14 10 5
```

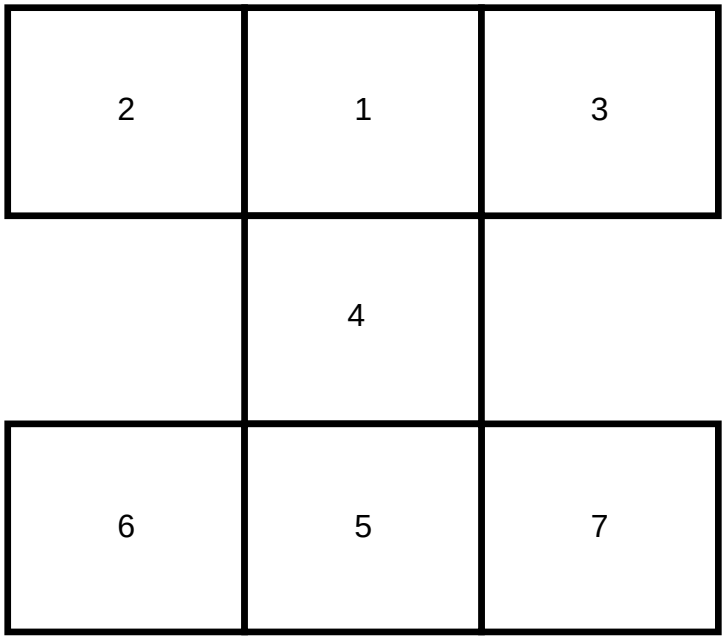
Figure (5.3) fichier de données de réf [3].

5. 2 L'effet des types d'éléments :

Pour montre la capacité de notre code et sa richesse en types d'éléments finis au choix de l'utilisateur, on propose ici quatre exemples de formes simples, à savoir : structure en forme de I, C, Carrée avec trou au centre et en forme de L. On a utilisé :

- l'élément triangulaire simple à 3 nœuds,
- l'élément quadrangulaire simple à 4 nœuds,
- l'élément triangulaire isoparamétrique à 6 nœuds,
- l'élément quadrangulaire isoparamétrique à 8 nœuds

a) Le premier exemple consiste en l'analyse d'une poutre de forme **I** en deux dimensions :



Nombre d'élément=7, nombre de nœud=38

On utilise l'élément triangulaire à 3 nœuds

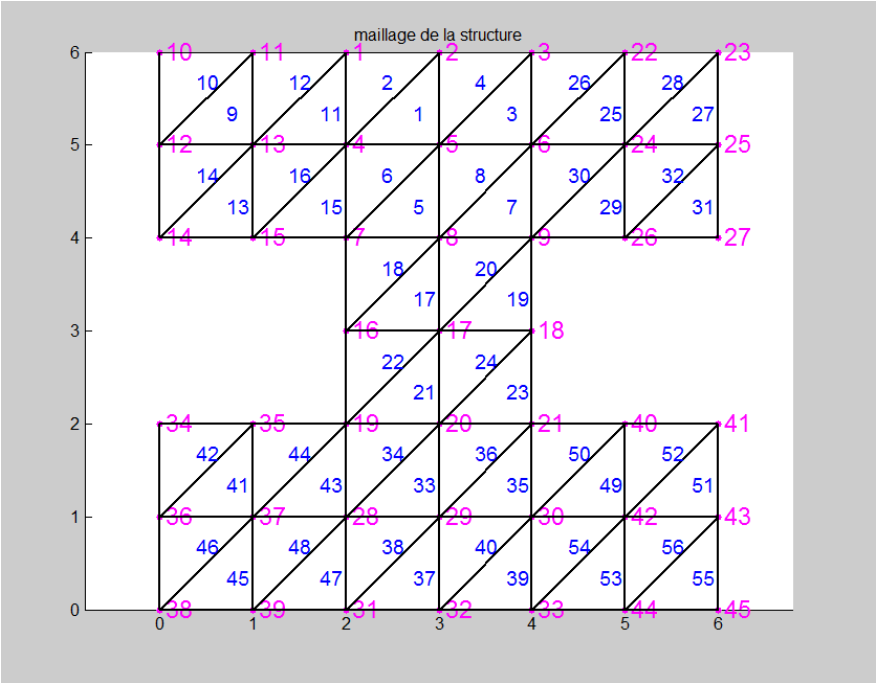


Figure 5. 4 : Maillage généré par l'élément linéaire à trois nœuds

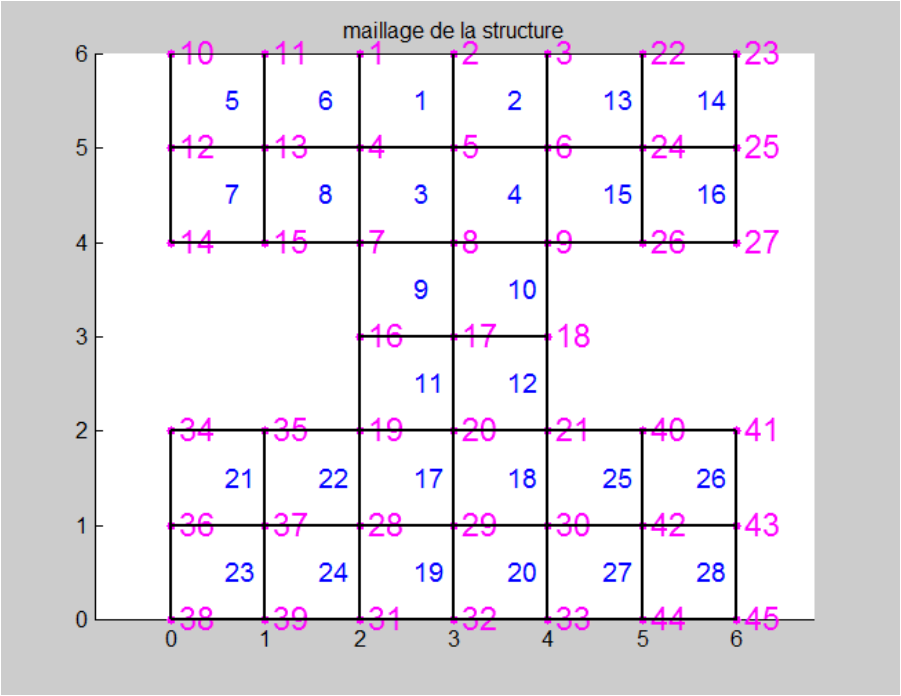


Figure 5. 5 : Maillage généré par l'élément linéaire à 4 nœuds

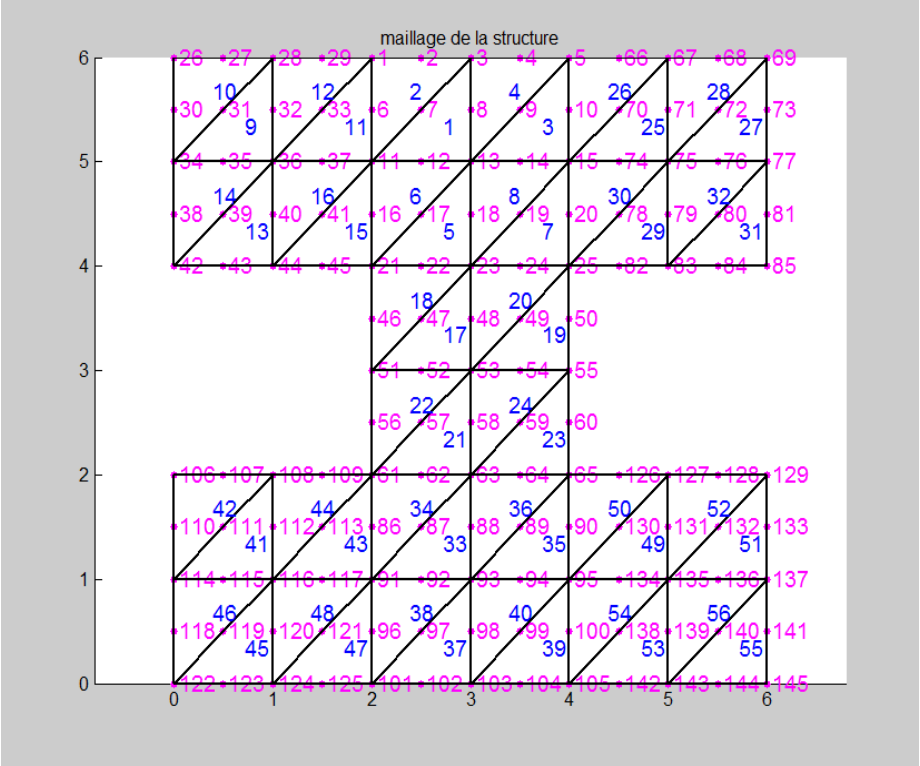


Figure 5. 6 : Maillage généré par l'élément quadratique à 6 nœuds

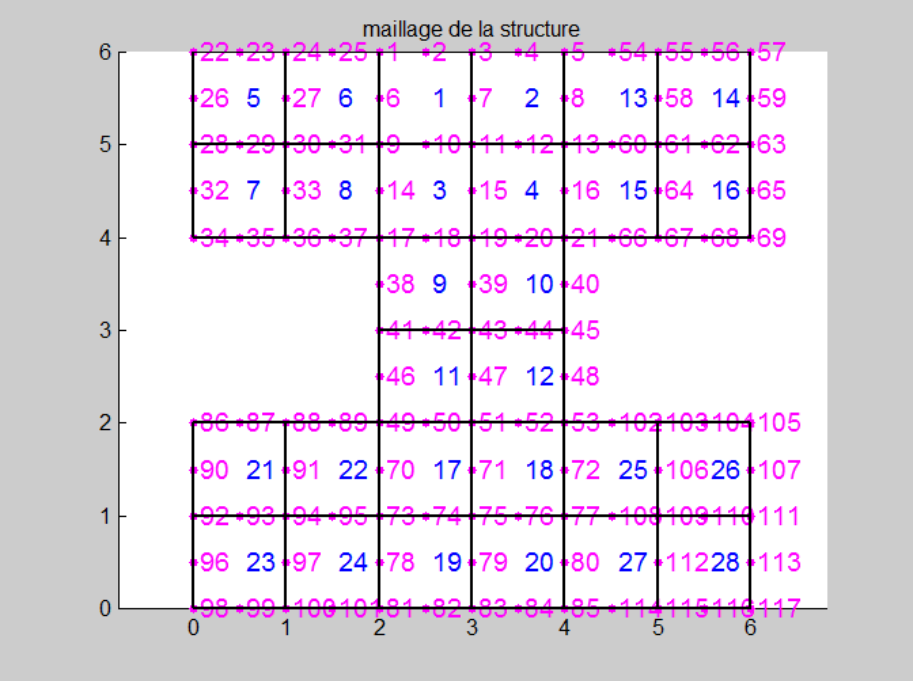
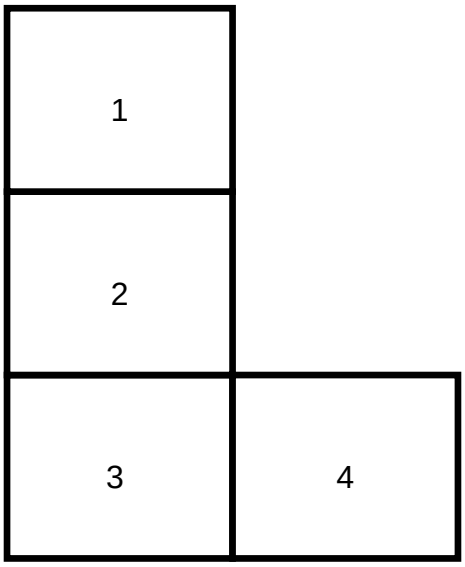


Figure 5. 7 : Maillage généré par l'élément quadratique à 8 nœuds

b) Le deuxième est un exemple d'une pièce mécanique de forme L :



Nombre d'élément=4, Nombre de nœud=23

On utilise l'élément triangulaire à 3 nœuds

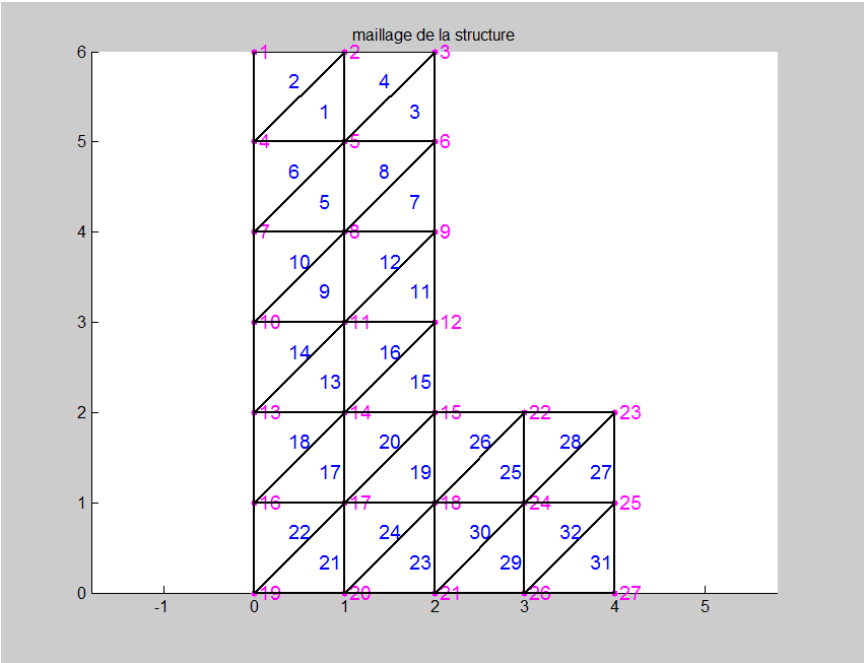


Figure 5. 8: Maillage généré par l'élément linéaire à 3 nœuds

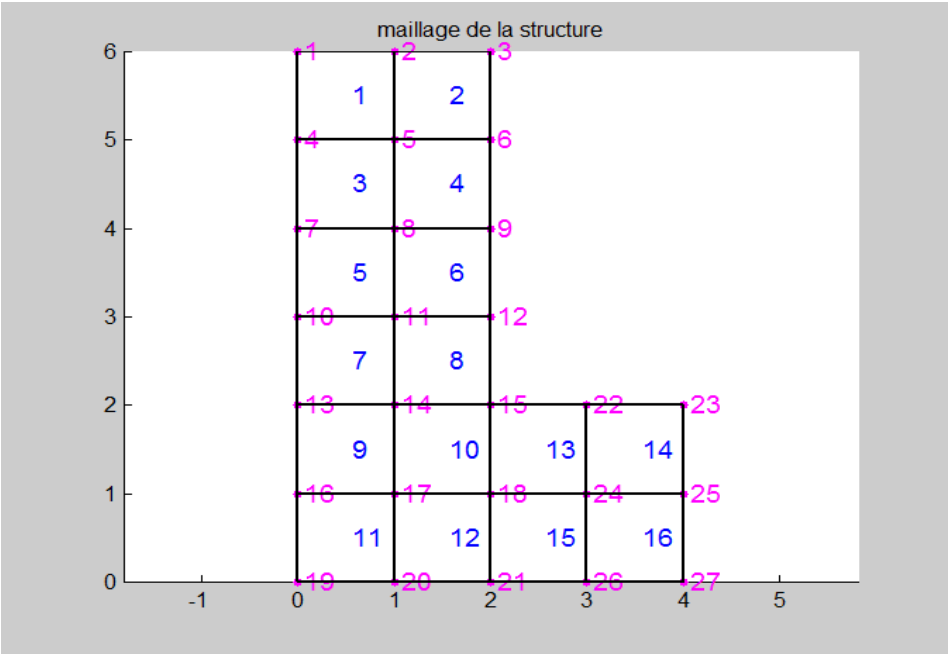


Figure 5. 9: Maillage généré par l'élément linéaire à 4 nœuds

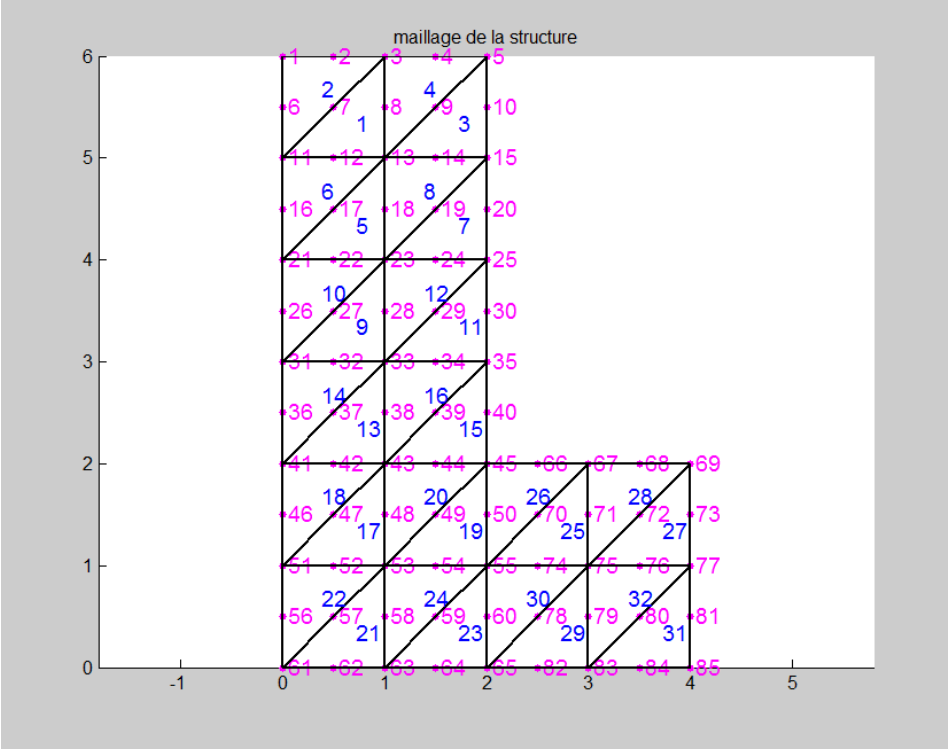


Figure 5. 10 : Maillage généré par l'élément quadratique à 6 nœuds

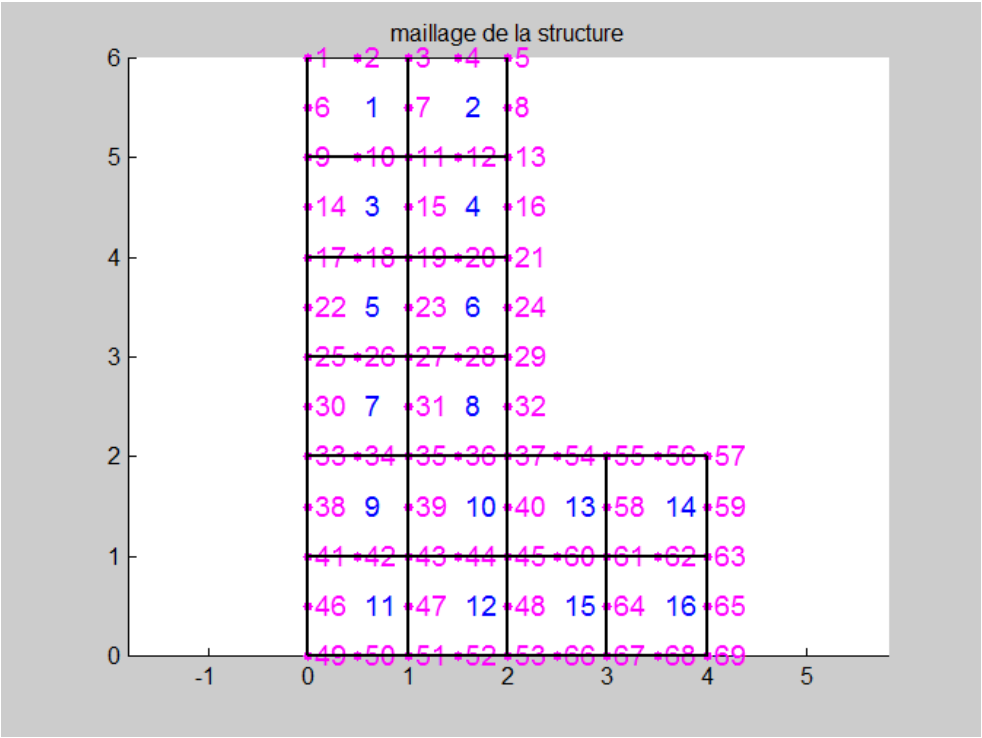
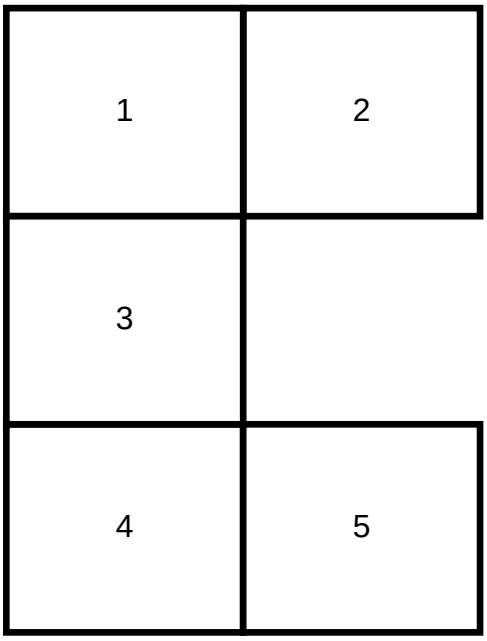


Figure 5. 11 : Maillage généré par l'élément quadratique à 8 nœuds

c) Un exemple d'une pièce mécanique de forme C :



Nombre d'élément=5, nombre de nœud=28

On utilise l'élément triangulaire à 3 nœuds

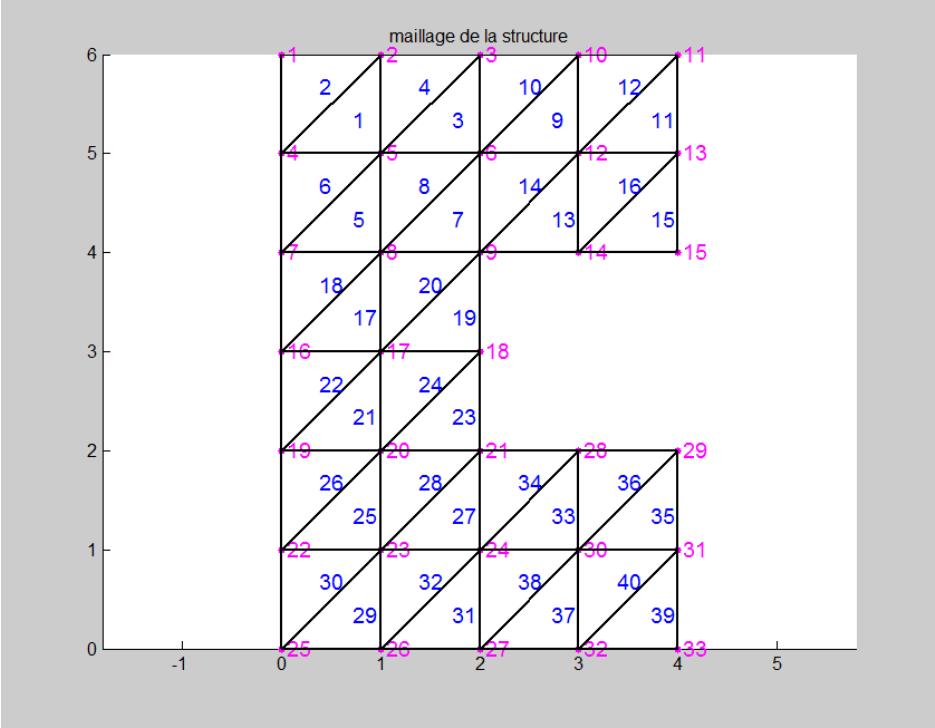


Figure 5. 12 : Maillage généré par l'élément linéaire à 3 nœuds

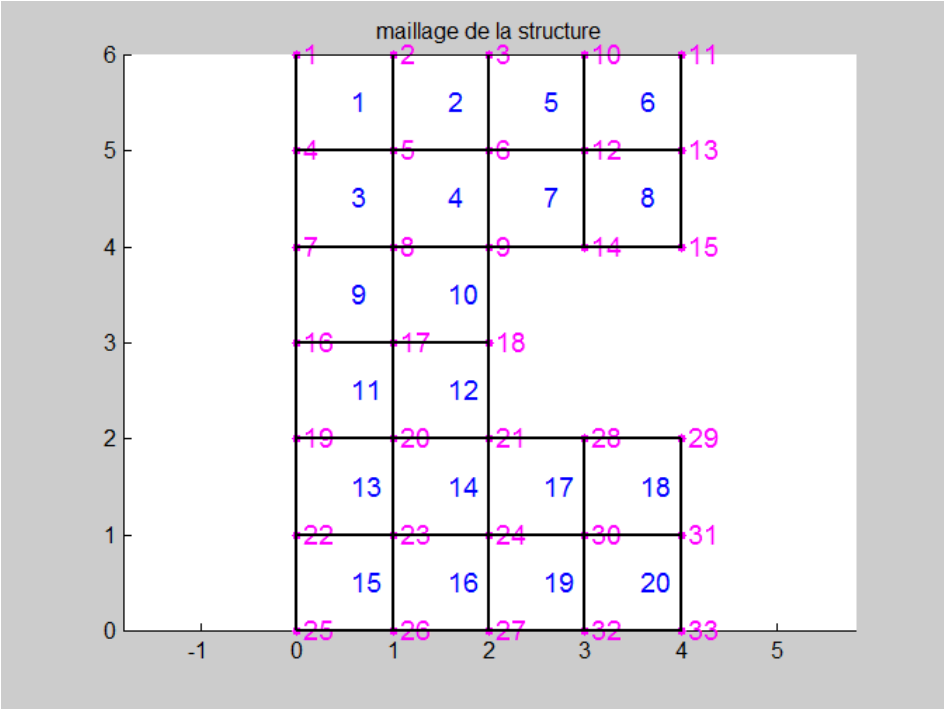


Figure 5. 13 : Maillage généré par l'élément linéaire à 4 nœuds

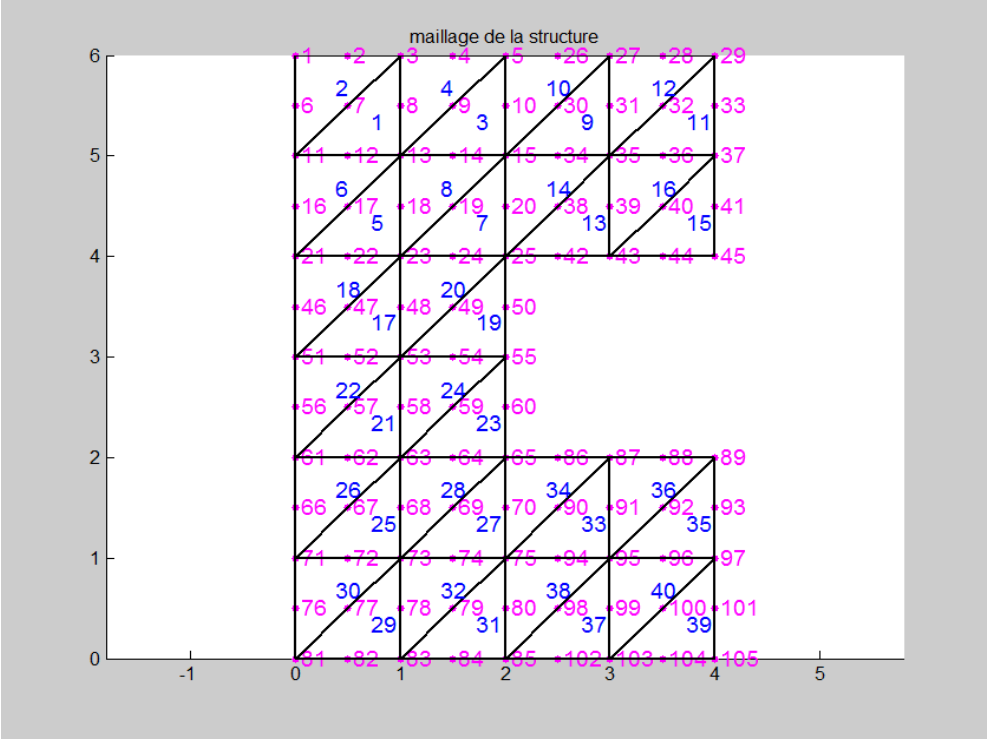


Figure 5. 14 : Maillage généré par l'élément linéaire à 6 nœuds

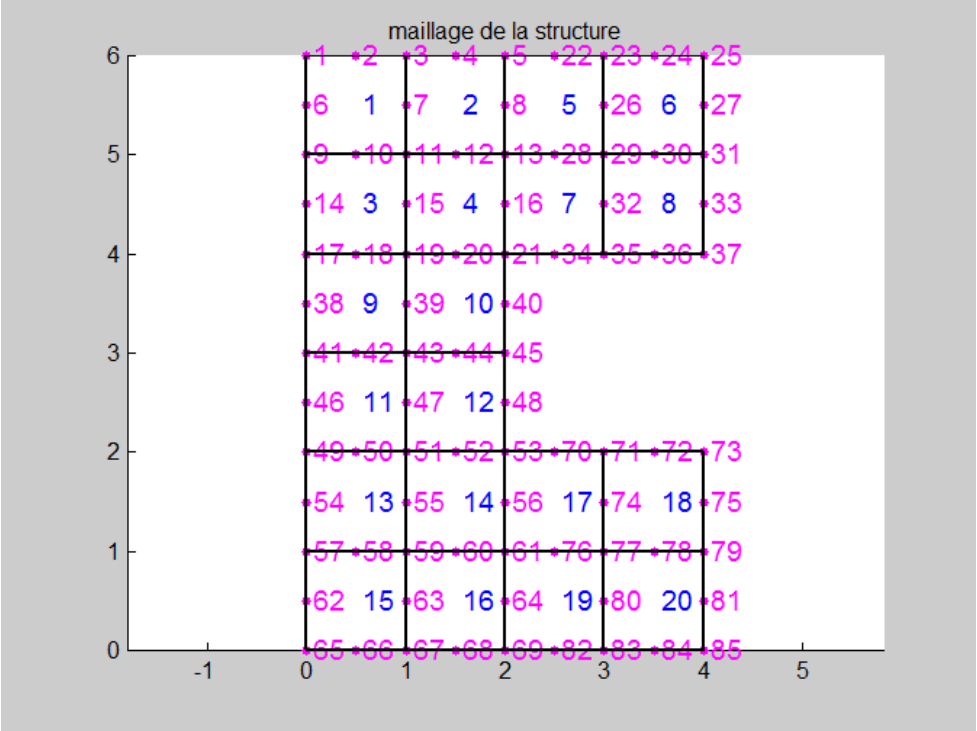


Figure 5. 15 : Maillage généré par l'élément quadratique à 8 nœuds

d) Un exemple d'une pièce mécanique de forme O :

2	1	3
4		5
6	7	8

Nombre d'élément=8, nombre de nœud=40

On utilise l'élément triangulaire à 3 nœuds

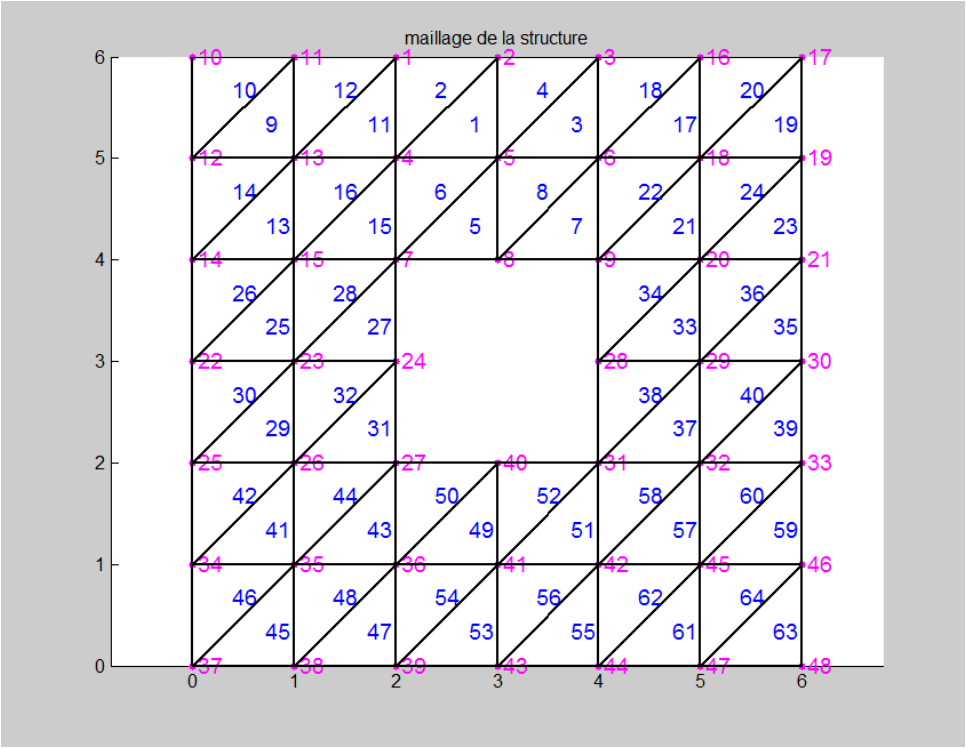


Figure 5. 16 : Maillage généré par l'élément linéaire à 3 nœuds

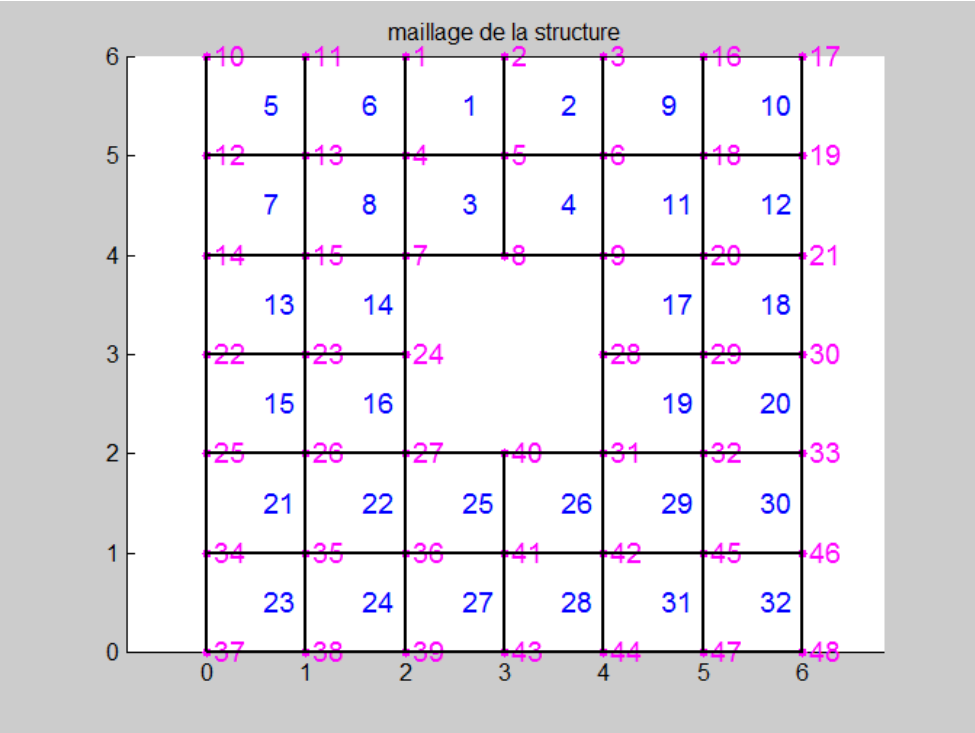


Figure 5. 17 : Maillage généré par l'élément linéaire à 4 nœuds

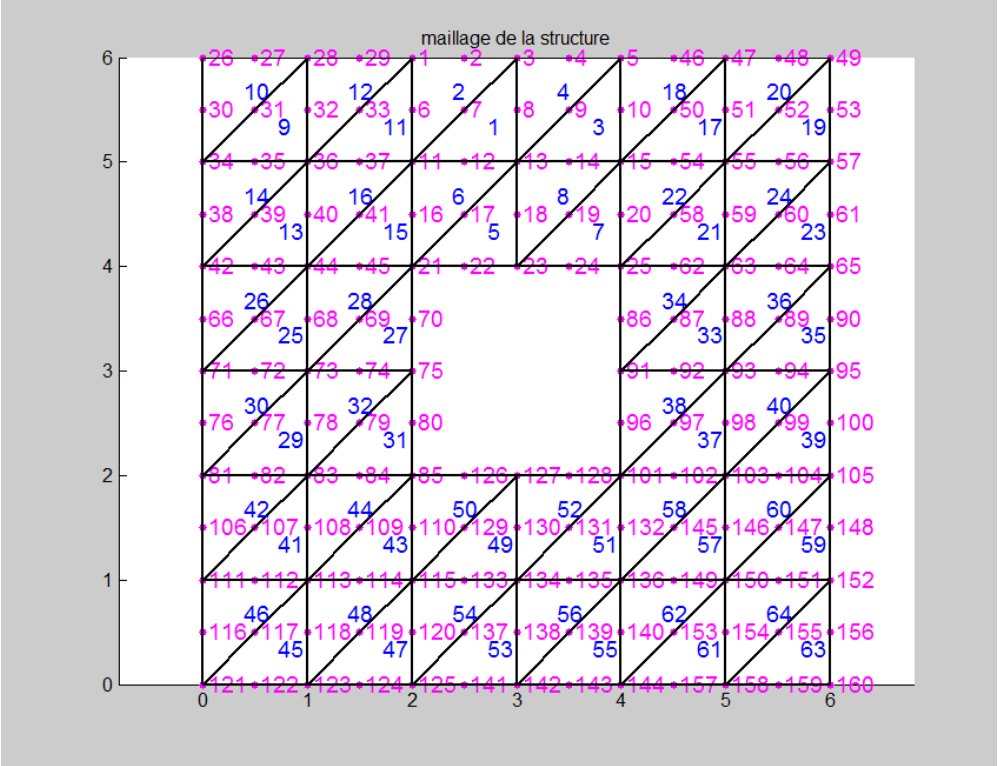


Figure 5. 18 : Maillage généré par l'élément quadratique à 6 nœuds

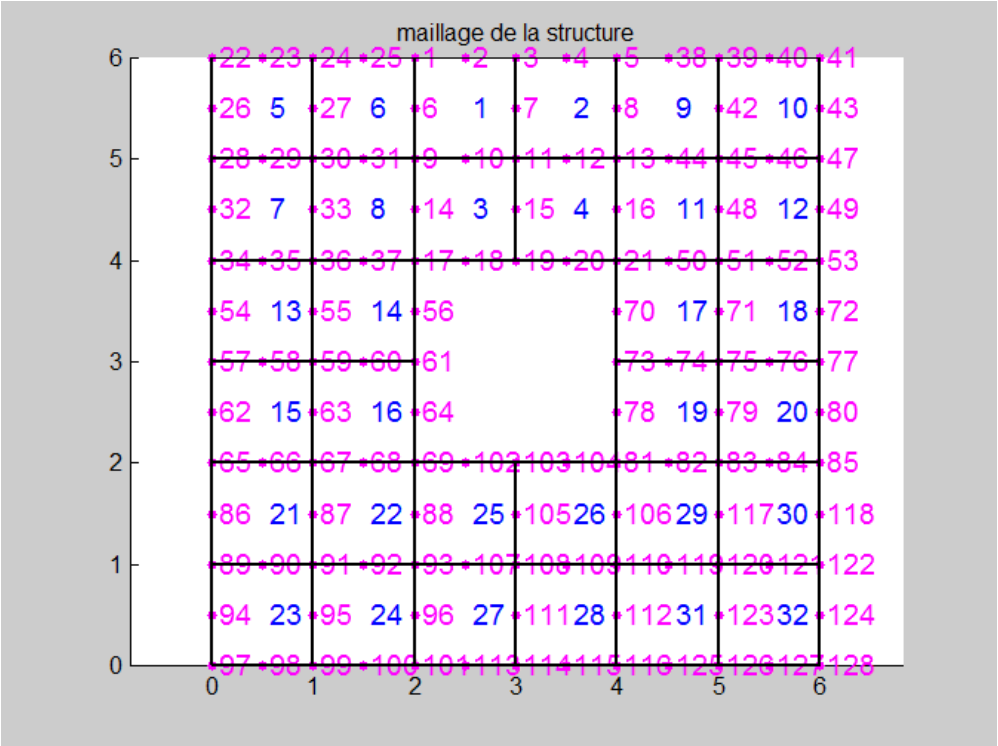
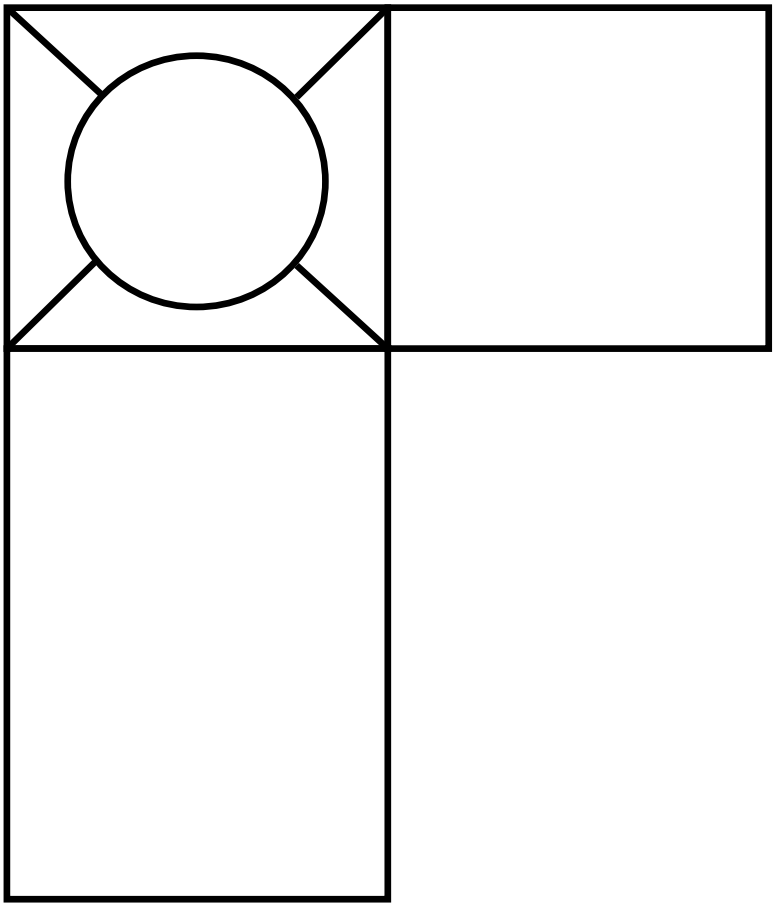


Figure 5. 19 : Maillage généré par l'élément quadratique à 8 nœuds

5. 3 Application à des géométries complexes et l'effet de raffinement de maillage :

On présente ici l'effet du raffinement de maillage à quelques structures complexes souvent rencontrés en génie mécanique.

a) Exemple 1 :



Nombre d'élément=6

Nombre de nœud= 30

On utilise l'élément triangulaire à 3 nœuds

Avec E =nombre d'élément

N =nombre de nœud

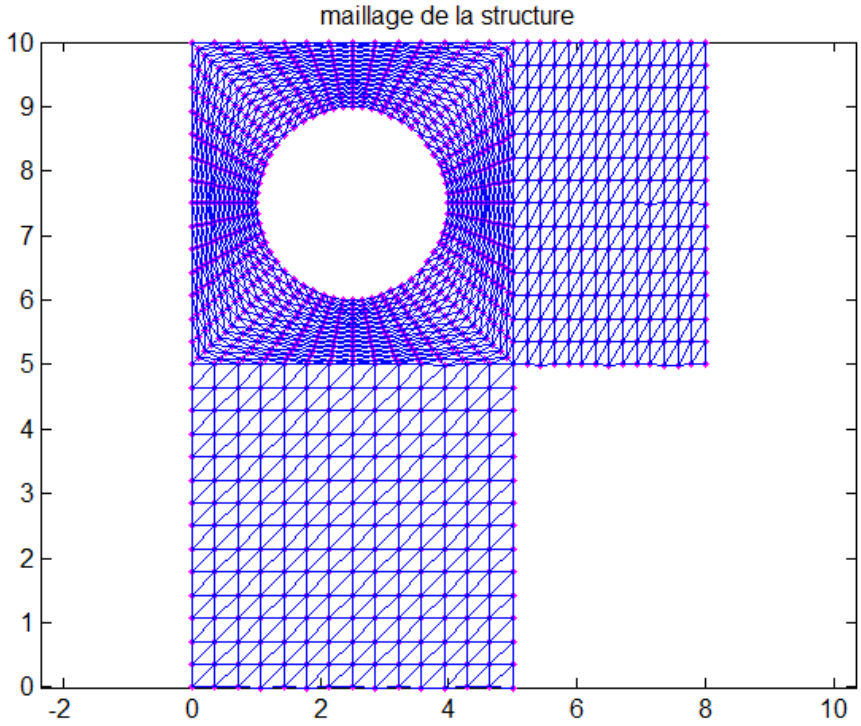


Figure 5. 20 : Maillage généré par l'élément linéaire à 3 nœuds

N= 1260, E= 2352

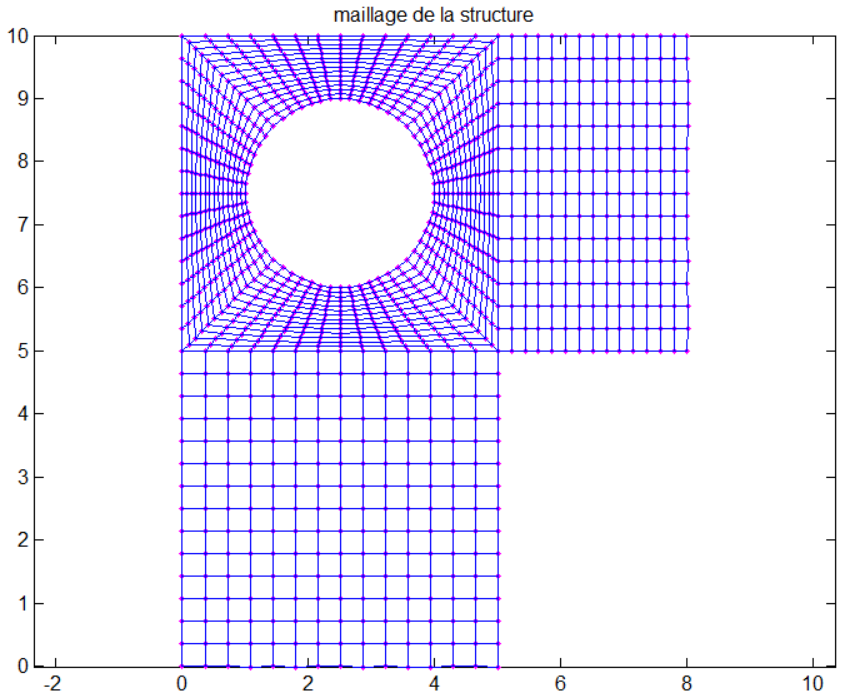


Figure 5. 21 : Maillage généré par l'élément linéaire à 4 nœuds avec raffinement

N= 1260, E= 1176

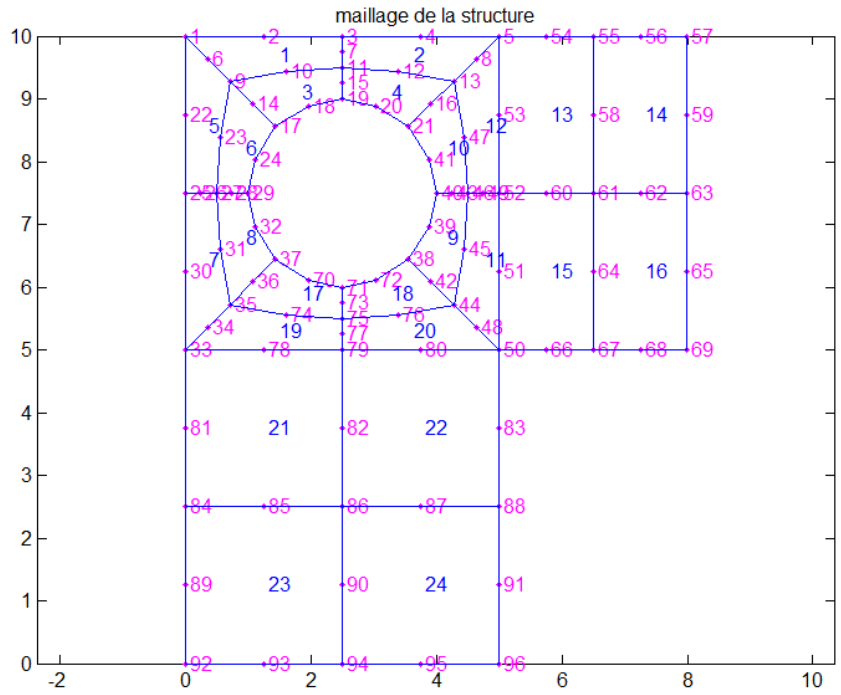


Figure 5. 22 : Maillage généré par l'élément quadratique à 8 nœuds

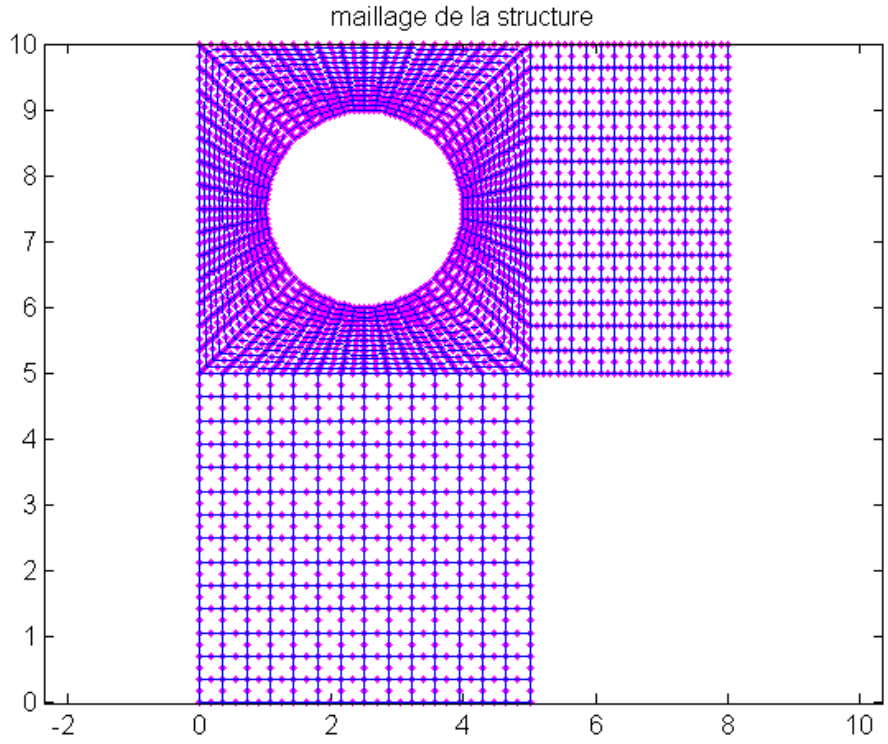
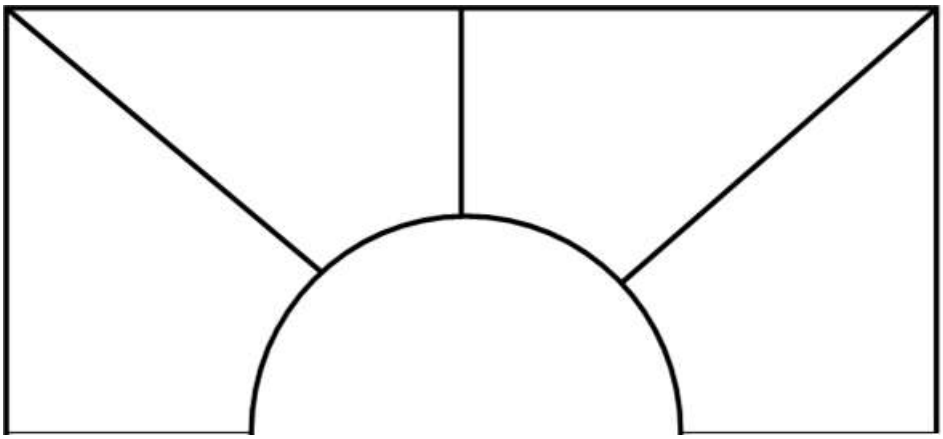


Figure 5. 23 : Maillage généré par l'élément quadratique à 8 nœuds avec raffinement

N= 3696, E= 1176

b) **Exemple 2 :** Un autre problème rencontré en génie mécanique



Nombre d'élément=4, Nombre de nœud= 23

En utilise l'élément triangulaire à 3 nœuds

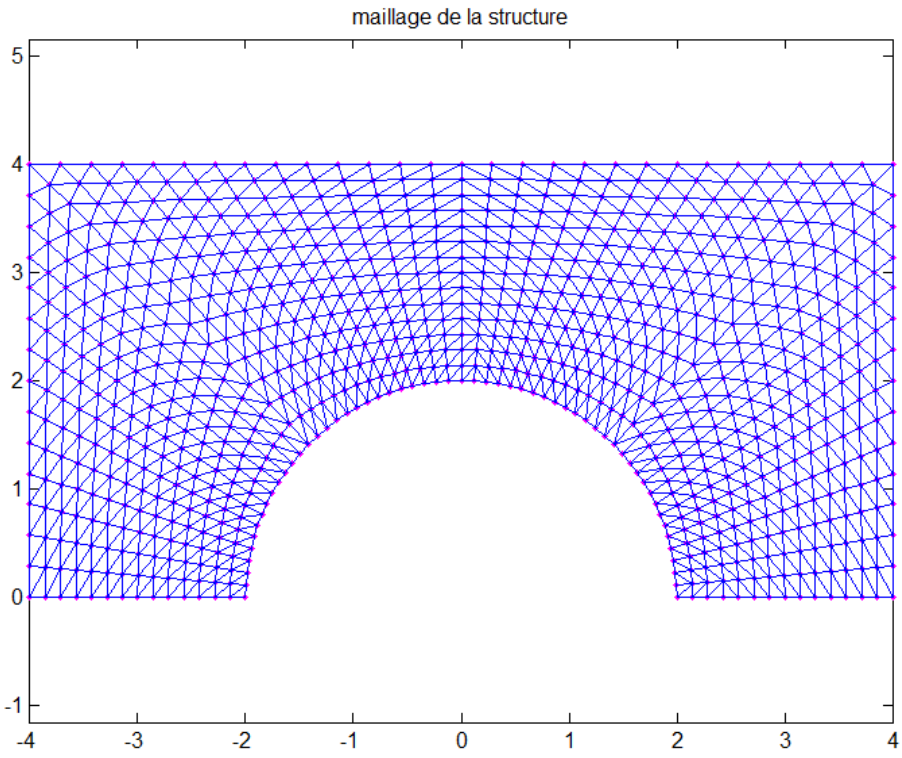


Figure 5. 24 : Maillage généré par l'élément linéaire à 3 nœuds avec raffinement

N= 855, E= 1568

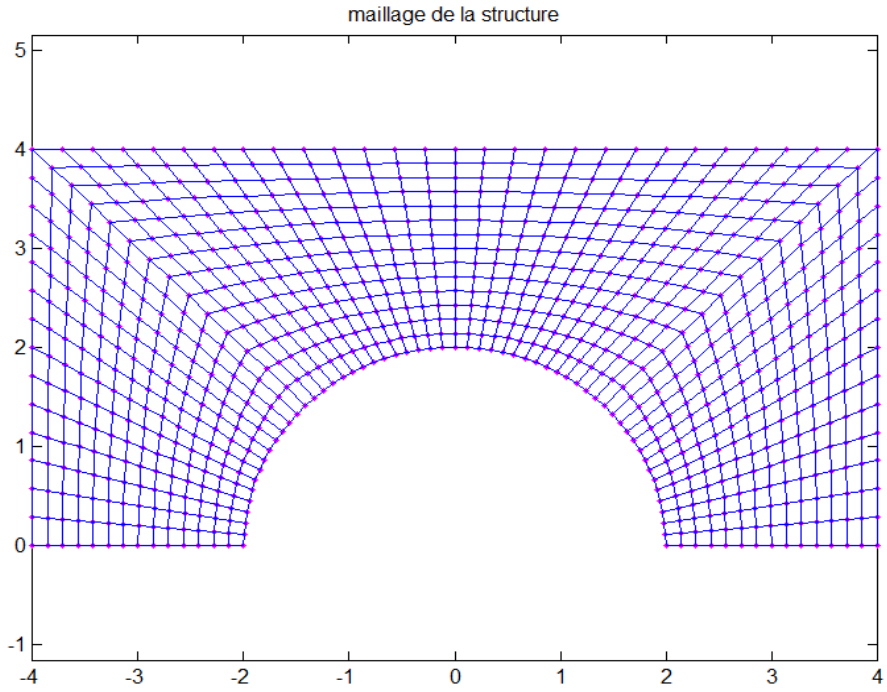


Figure 5. 25 : Maillage généré par l'élément linéaire à 4 nœuds avec raffinement

N= 855, E= 784

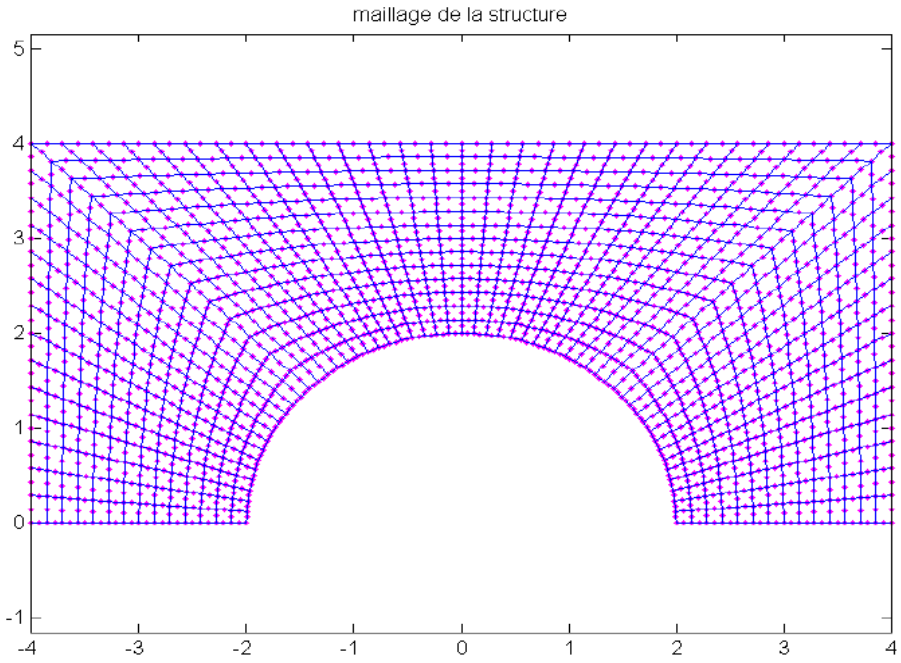


Figure 5. 26 : Maillage généré par l'élément quadratique a 8 nœuds avec raffinement

N= 2493, E= 784

c) Exemple 3 :

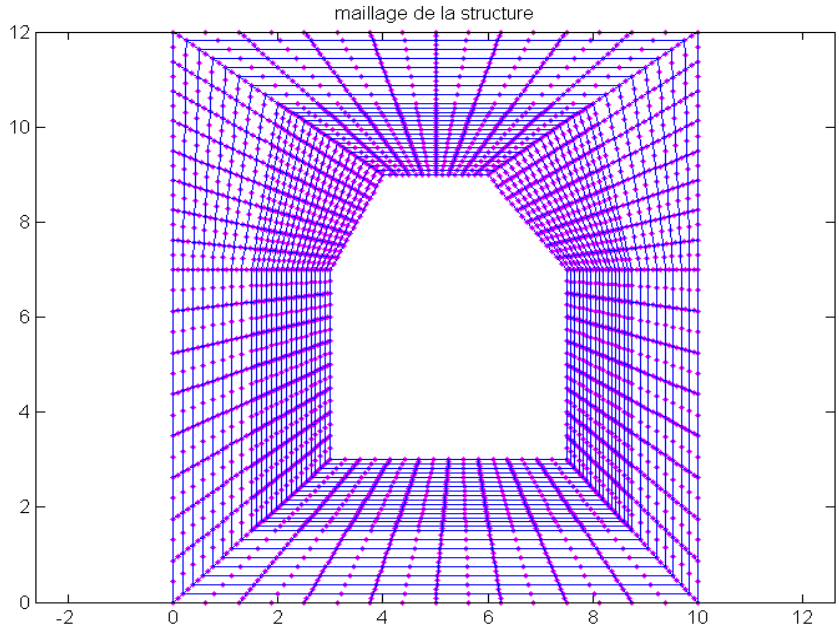


Figure 5. 27 : Maillage généré par l'élément quadratique à 8 nœuds avec raffinement

Nombre d'élément=1152 Nombre de nœud= 3552

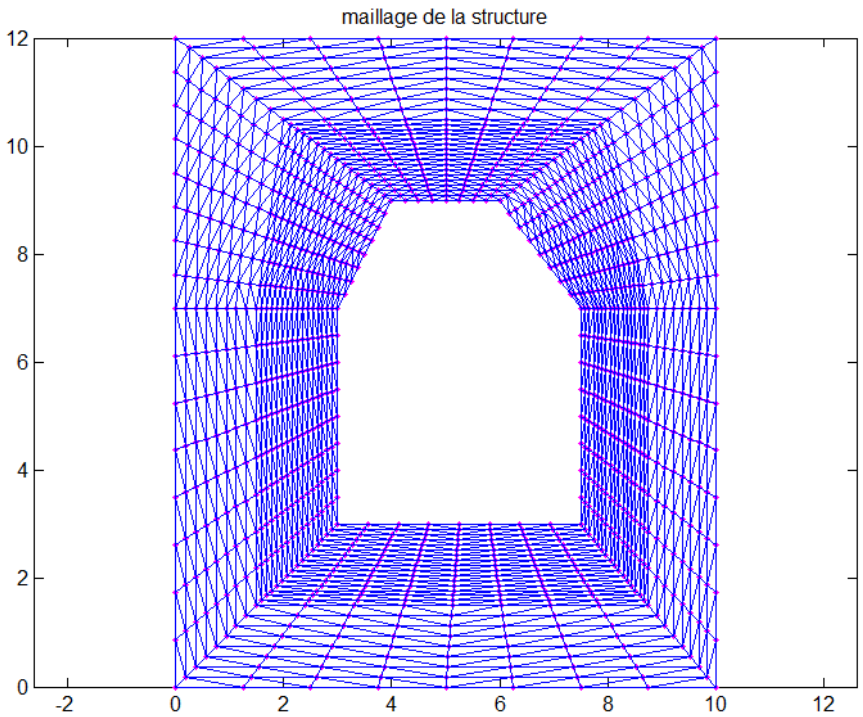


Figure 5. 28 : Maillage généré par l'élément linéaire à 3 nœuds avec raffinement

Nombre de nœud= 1200 Nombre d'élément=2304

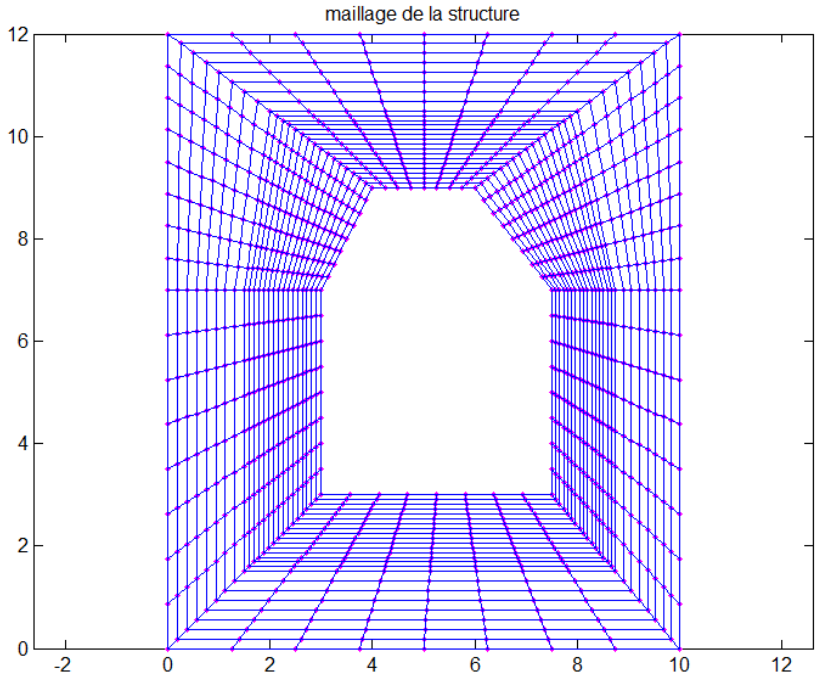


Figure 5. 29 : Maillage généré par l'élément linéaire a 4 nœuds avec raffinement (nombre de nœud= 1200 , nombre d'élément=1152)

d) Exemple 4 :

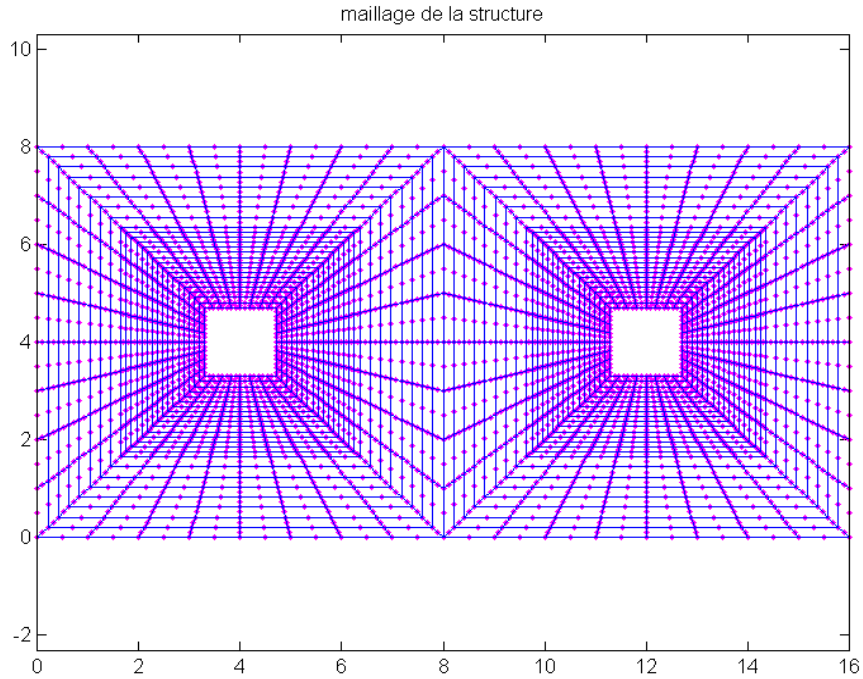


Figure 5. 30 : Maillage généré par l'élément quadratique a 8 nœuds avec raffinement (nombre de nœud= 4719 , nombre d'élément=1536)

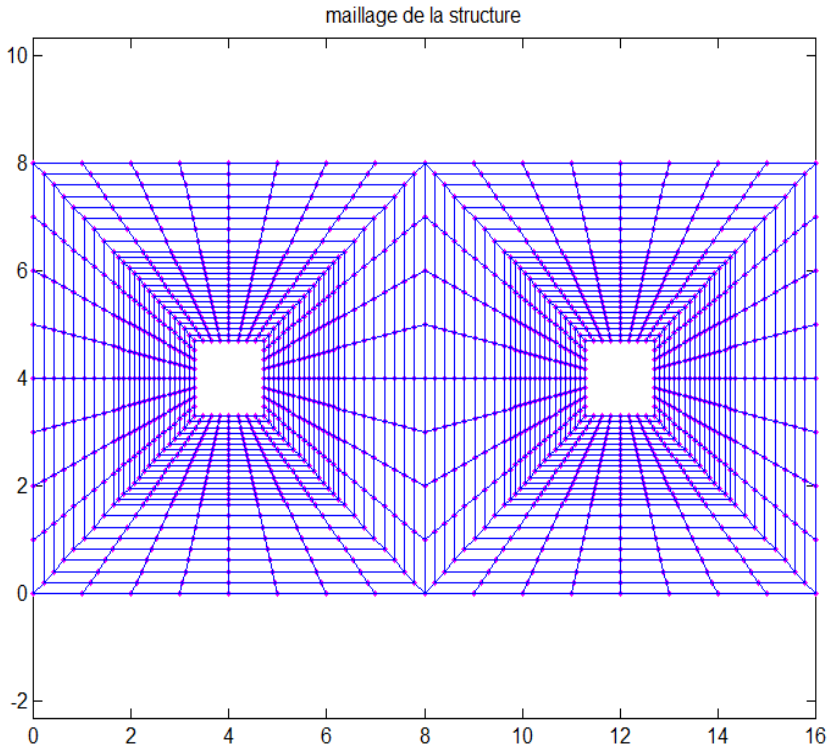


Figure 5. 31 : Maillage généré par l'élément linéaire à 4 nœuds avec raffinement (nombre de nœud= 1591 , nombre d'élément=1536)

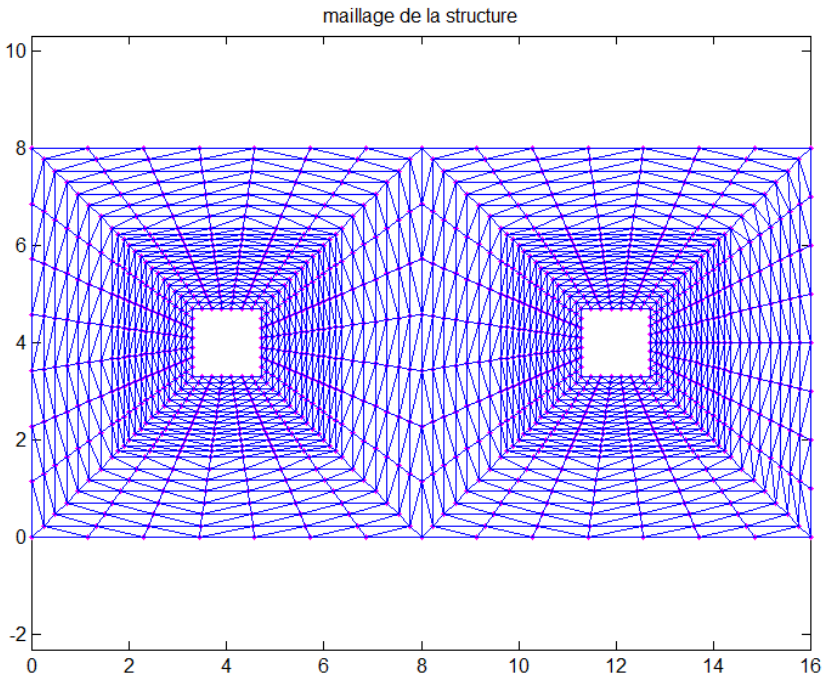


Figure 5. 32 : Maillage généré par l'élément linéaire à 3 nœuds avec raffinement (nombre de nœud= 1253 , nombre d'élément=2410)

e) Exemple 5 :

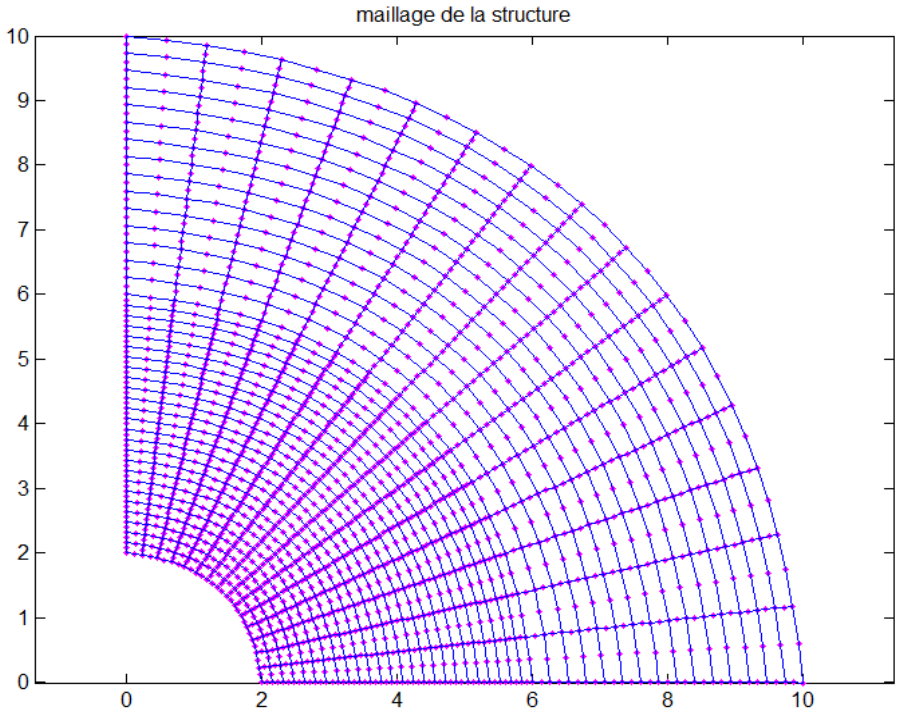


Figure 5. 33 : Maillage généré par l'élément quadratique a 8 nœuds avec raffinement (nombre de nœud= 1911 , nombre d'élément=600)

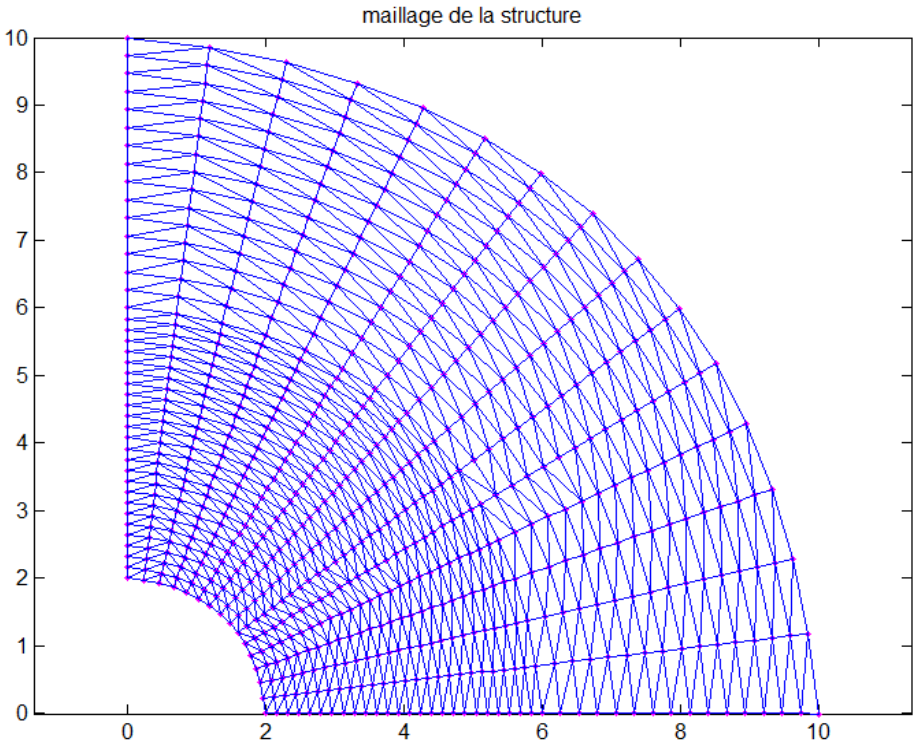


Figure 5. 34 : Maillage généré par l'élément linéaire a 3 nœuds avec raffinement (nombre de nœud= 656 , nombre d'élément=1200)

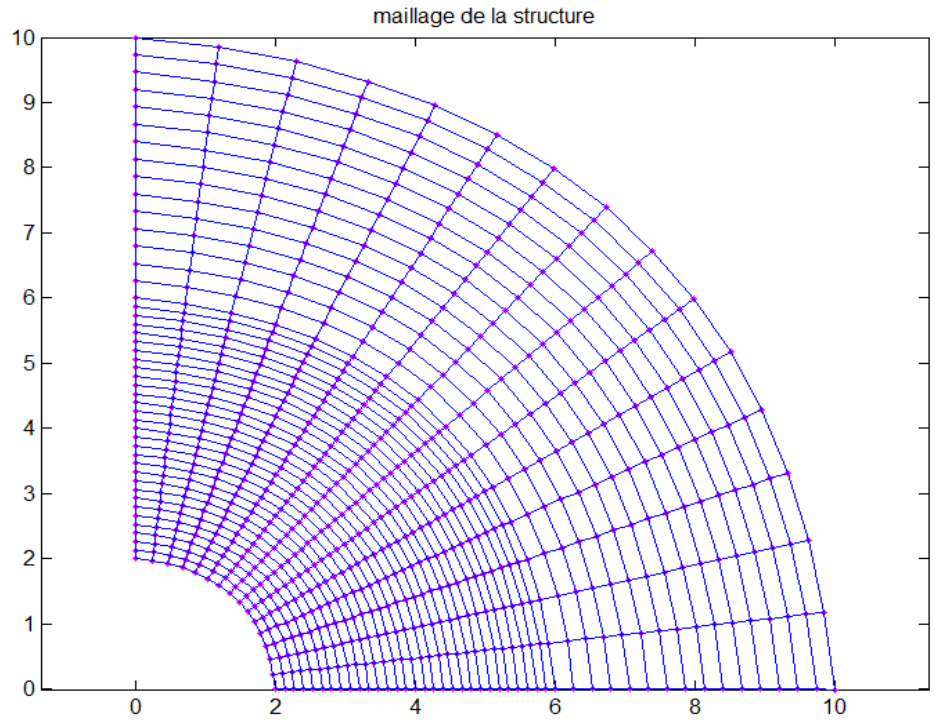


Figure 5. 35 : Maillage généré par l'élément linéaire à 4 nœuds avec raffinement (nombre de nœud= 736 , nombre d'élément=675)

f) Exemple 6 :

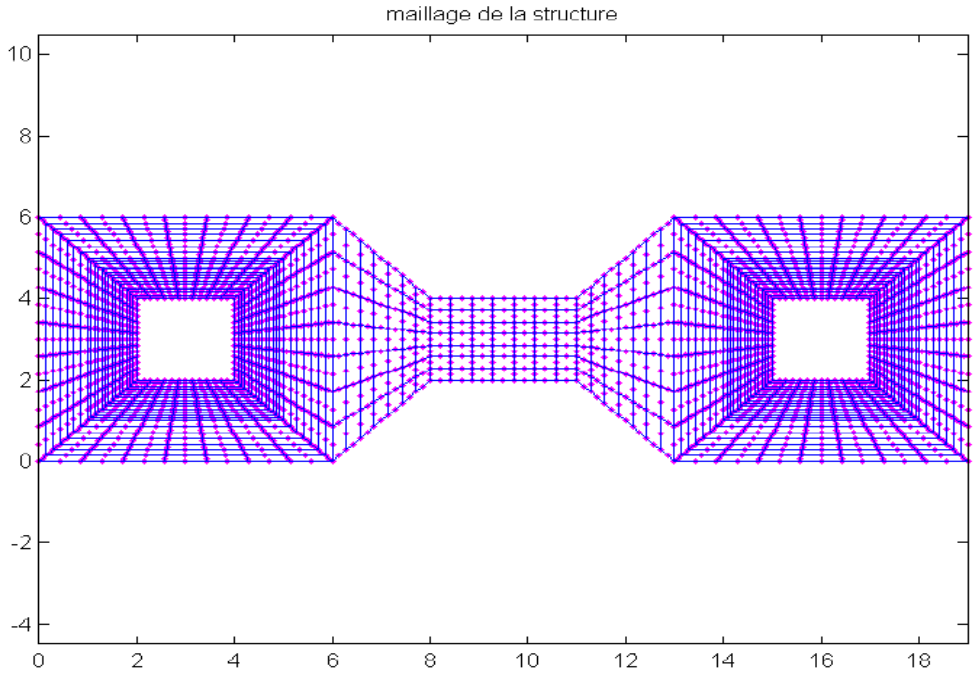


Figure 5. 36 : Maillage généré par l'élément quadratique à 8 nœuds avec raffinement (nombre de nœud= 4108 , nombre d'élément=1323)

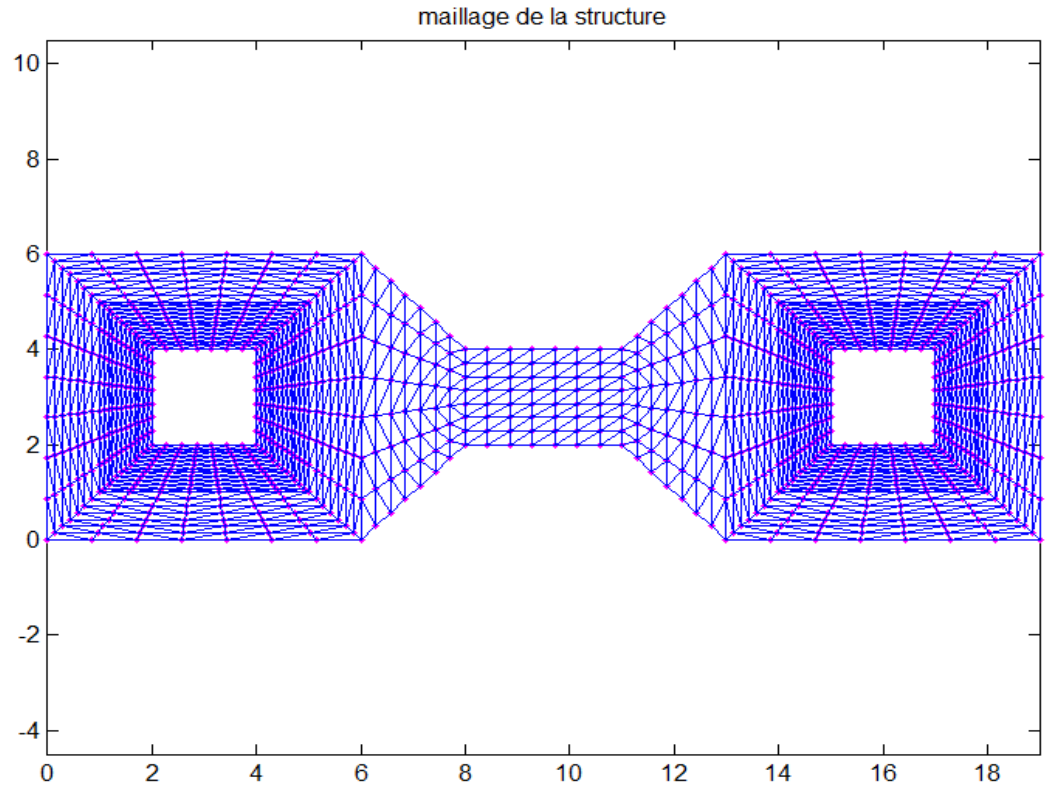


Figure 5. 37 : Maillage généré par l'élément linéaire a 3 nœuds avec raffinement (nombre de nœud= 1280 , nombre d'élément=2422)

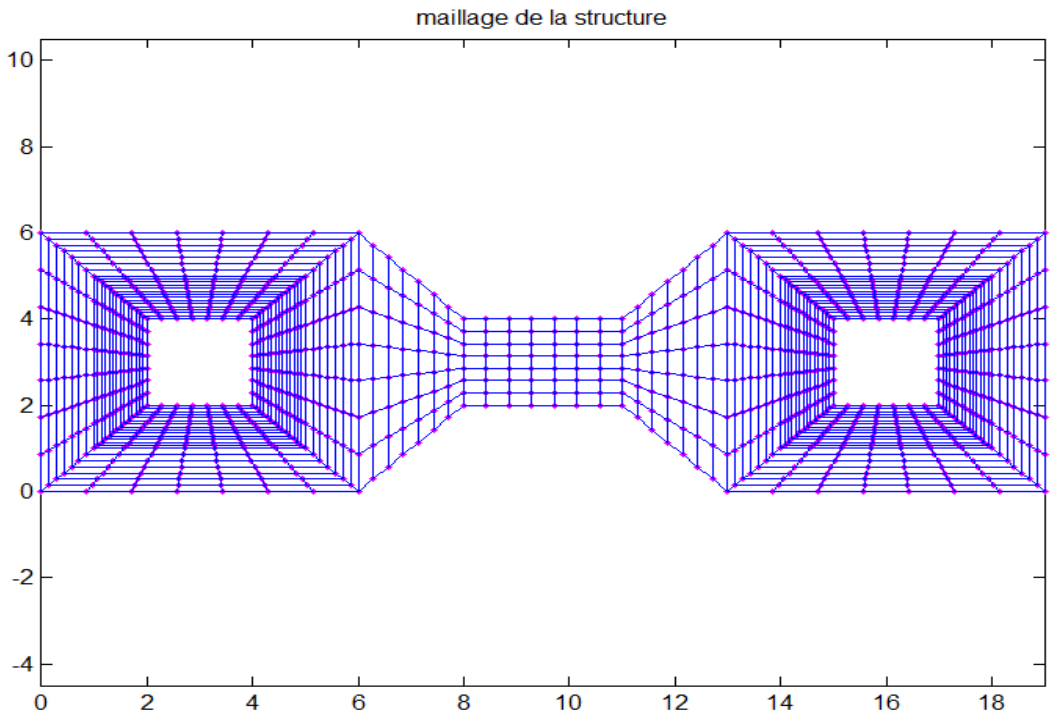


Figure 5. 38 : Maillage généré par l'élément linéaire a 4 nœuds avec raffinement (nombre de nœud= 1392 , nombre d'élément=1323)

5. 5 Application à des problèmes d'élasticités planes :

5. 5.1 Introduction :

Dans cette partie, l'approche de contrainte est étendue à étudier la performance de éléments triangulaires à 3 nœuds ou quadrangulaires à 4 nœuds est étudiée en l'appliquant à la solution de 3 problèmes dans l'élasticité plane. Le but est de simuler le fléchissement d'une poutre dont l'une des extrémités est encastrée soumise à une charge constante. Et l'autre une poutre en acier encastrée soumise à une charge répartie constante et le troisième cas une poutre supportée aux extrémités sous l'effet d'une charge constante.

Ces 3 problèmes sont simple possède une solution analytique issue de la théorie des poutres. Cela nous permettra de comparer la véracité des résultats numériques obtenus pour différents maillages.

5. 5.1.1 Etude de la flexion d'une poutre en acier encastrée soumise à une charge répartie constante :

On considère un cas pratique (voir Figure 5. 36 ci-dessous) de poutre en acier encastrée soumise à une charge répartie constante p = 100 N/mm. On considère également que :

E=210000 MPa

$\nu = 0,28$

Epaisseur e = 10 mm

Longueur L = 90 mm

Hauteur H = 30 mm

On fait varier deux types de paramètres du maillage :

- le type d'éléments :

  - triangles à 3 nœuds ;

- quadrangles à 4 nœuds ;

- la densité des éléments.

Différents maillages (suivant x, suivant y) :

triangle à 3 nœuds	Quadrangle à 4 nœuds
a) maillage 1 x 3; b) maillage 2 x 6; c) maillage 3 x 9; d) maillage 3 x 12; e) maillage 5 x 15; f) maillage 6 x 18; g) maillage 7 x 21; h) maillage 8 x 24; i) maillage 9 x 27;	a) maillage 1 x 3; b) maillage 2 x 6; c) maillage 3 x 9; d) maillage 3 x 12; e) maillage 5 x 15; f) maillage 6 x 18; g) maillage 7 x 21; h) maillage 8 x 24; i) maillage 9 x 27;

Table 5. 1: résultats des déplacements d'une poutre en acier encastrée soumise à une charge répartie constante qui utilise l'élément quadrangle à 4 nœuds

Nature des éléments	Maillage	Nb nœuds	Nb éléments	Valeur absolue du déplacement suivant y en mm $\times 10^{-3}$	La solution analytique correspond à $\times 10^{-3}$	Erreur de la solution
Quadrangle à 4 nœuds	a	**8**	3	0134.44	197	31.75%
	b	**21**	12	0171.39	197	13.00%
	c	**40**	27	0182.68	197	7.27%
	d	**65**	64	0187.26	197	4.94%
	e	**96**	75	0189.55	197	3.78%
	f	**133**	108	0190.85	197	3.12%
	g	**176**	147	0191.66	197	2.71%
	h	**225**	192	0192.21	197	2.43%
	i	**280**	243	0192.59	197	2.23%

Table 5. 2: les résultats de déplacements d'une poutre en acier encastrée soumise à une charge répartie constante qui utilise l'élément triangle à 3 nœuds

Nature des éléments	Maillage	Nb nœuds	Nb éléments	Valeur absolue du déplacement suivant y en mm $\times 10^{-3}$	La solution analytique correspond à $\times 10^{-3}$	Erreur de la solution
Triangle à 3 nœuds	a	**8**	6	0056.019	197	71.563%
	b	**21**	24	0109.86	197	44.385%
	c	**40**	54	0142.63	197	27.598%
	d	**65**	96	0160.59	197	18.482%
	e	**96**	150	0170.88	197	13.258%
	f	**133**	216	0177.16	197	10.071%
	g	**176**	294	0181.24	197	7.999%
	h	**225**	384	0184.01	197	6.593%
	i	**280**	486	0185.98	197	5.593%

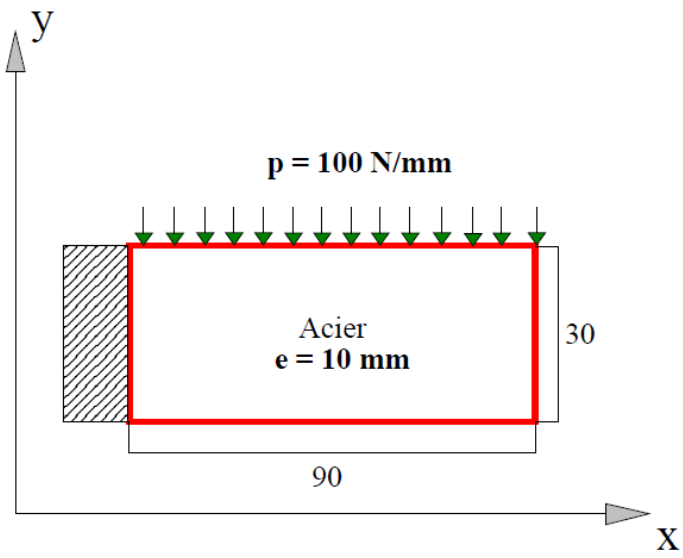


Figure 5. 39 : Poutre encastrée avec charge répartie constante

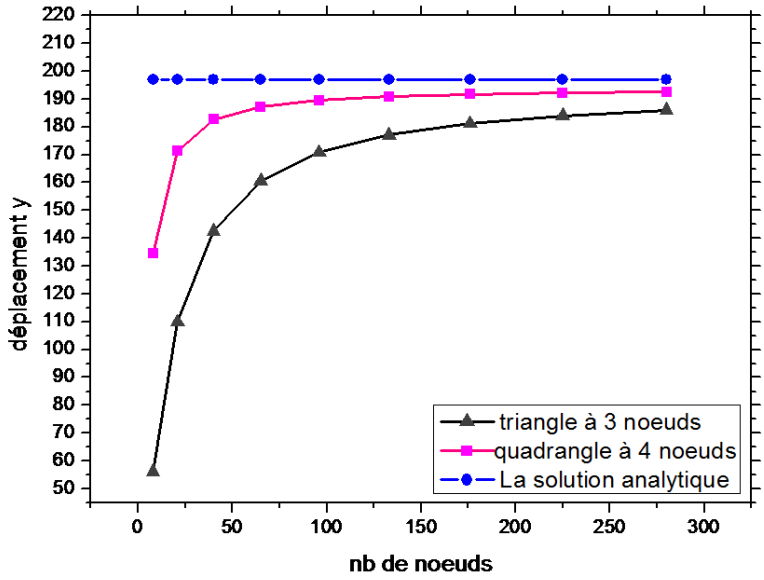


Figure 5. 40 : la variation de déplacement y en fonction de nb de nœuds

La figure (5.40) montre que le triangle à 3 nœuds est très peu performant. Même lorsque le nombre de nœuds augmente, la solution ne tend que très lentement vers la solution exacte.

Ainsi, il ne sert à rien de se servir d'élément triangulaire à 3 nœuds en élevant inconsidérément le nombre de nœuds.

L'élément quadrangle à 4 nœuds est légèrement plus performant que le triangle à 3 nœuds. La solution est rapidement approchée, même avec un faible nombre de nœuds.

On peut donc conclure de cette petite étude que :

- le choix du type d'élément est très important ;

- en flexion, triangle à 3 nœuds et quadrangle à 4 sont des éléments sont économiques mais n’est pas performants.

En appliquant la formule utilisée dans la Résistance des matériaux, on aura:

$y_{max} = \frac{PL^4}{8EI} + \alpha \frac{PL^2}{2AG}$ et vaut y_{max}=0.197 mm

Avec

$\alpha = \frac{3}{2}$ section rectangulaire ; $\alpha = \frac{4}{3}$ section circulaire

A = aire de la section droite

I_Z= moment quadratique suivant *z*

E = module de YOUNG

$G = \frac{E}{2(1+\nu)}$

ν = coefficient de Poisson

5. 5.1.2 Etude de la flexion d'une poutre supportée aux extrémités sous l'effet d'une charge :

Considérons une poutre en acier supportée aux extrémités (voir figure 5. 38 ci-dessous). On suppose que cette poutre est uniforme et que toutes ses sections transversales sont identiques. Sous l'effet d'une charge constante F = 100000 N, la poutre se déforme. On considère également que :

E=210000 MPa
ν = 0,28
Epaisseur e = 10 mm
Longueur L = 90 mm
Hauteur H = 30 mm
On fait varier deux types de paramètres du maillage :
- le type d'éléments :
  - triangles à 3 nœuds ;
  - quadrangles à 4 nœuds ;

- la densité des éléments.

Différents maillages (suivant x, suivant y) :

triangle à 3 nœuds	Quadrangle à 4 nœuds
a) maillage 1x4 ; b) maillage 2x8 ; c) maillage 3x12; d) maillage 4x16; e) maillage 5x 20; f) maillage 6x 24 ; g) maillage 7x28 ; h) maillage 8x32 ; i) maillage 9x36 ; j) maillage 10x40 ;	a) maillage 1x4 ; b) maillage 2x8 ; c) maillage 3x12; d) maillage 4x16; e) maillage 5x 20; f) maillage 6x24 ; g) maillage 7x28 ; h) maillage 8x32 ; i) maillage 9x36 ; j) maillage 10x40 ;

Table 5. 3: résultats de déplacements d'une poutre supportée aux extrémités sous l'effet d'une charge constante qui utilise l'élément quadrangle à 4 nœuds

Nature des éléments	Maillage	Nb nœuds	Nb éléments	Valeur absolue du déplacement suivant y en mm $\times 10^{-3}$	La solution analytique correspond à $\times 10^{-3}$	Erreur de la solution
Quadrangle à 4 nœuds	a	**10**	4	0221.94	0431.14	48.52%
	b	**27**	16	0279.30	0431.14	35.21%
	c	**52**	36	0314.95	0431.14	26.94%
	d	**85**	64	0338.15	0431.14	21.56%
	e	**126**	100	0355.09	0431.14	17.63%
	f	**175**	144	0368.40	0431.14	14.55%
	g	**232**	196	0379.36	0431.14	12.01%
	h	**297**	256	0388.67	0431.14	9.85%
	i	**370**	324	0396.77	0431.14	7.97%
	j	**451**	400	0403.93	0431.14	6.31%

Table 5. 4 résultats de déplacements d'une poutre supportée aux extrémités sous l'effet d'une charge constante qui utilise l'élément triangle à 3 nœuds

Nature des éléments	Maillage	Nb nœuds	Nb éléments	Valeur absolue du déplacement suivant y en mm $\times 10^{-3}$	La solution analytique correspond à $\times 10^{-3}$	Erreur de la solution
Triangle à 3 nœuds	a	**10**	8	0161.83	0431.14	62.46%
	b	**27**	32	0227.00	0431.14	47.34%
	c	**52**	72	0267.28	0431.14	38.00%
	d	**85**	128	0294.40	0431.14	31.71%
	e	**126**	200	0314.15	0431.14	27.13%
	f	**175**	288	0329.43	0431.14	23.59%
	g	**232**	392	0341.79	0431.14	20.72%
	h	**297**	512	0352.13	0431.14	18.32%
	i	**370**	648	0361.01	0431.14	16.26%
	j	**451**	800	0368.78	0431.14	14.46%

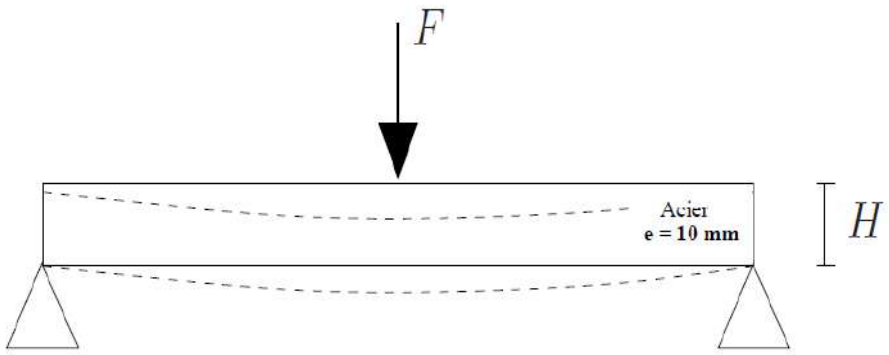


Figure 5. 41: poutre supportée aux extrémités sous l'effet d'une charge constante

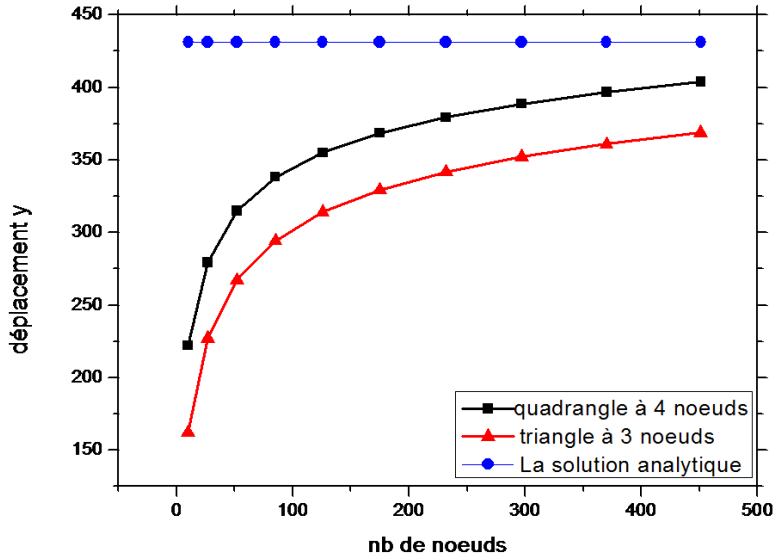


Figure 5. 42 : la variation de déplacement y en fonction de nb de nœuds.

La figure (5.42) montre que le triangle à 3 nœuds est très peu performant. Même lorsque le nombre de nœuds augmente, la solution ne tend que très lentement vers la solution exacte.

Ainsi, il ne sert à rien de se servir d'élément triangulaire à 3 nœuds en élevant inconsidérément le nombre de nœuds.

L'élément quadrangle à 4 nœuds est légèrement plus performant que le triangle à 3 nœuds. La solution est rapidement approchée, même avec un faible nombre de nœuds.

On peut donc conclure de cette petite étude que :

- le choix du type d'élément est très important ;

- en flexion, triangle à 3 nœuds et quadrangle à 4 sont des éléments sont économiques mais n'est pas performants.

Solution analytique :

En appliquant la formule utilisée dans la Résistance des matériaux, on aura:

$$y_{max} = \frac{FL^3}{48EI_Z} + \frac{FL^2h^2}{40GI_Z}$$ et vaut y_{max} =0.43114 mm

I_Z= moment quadratique suivant *z*

E = module de YOUNG

$$G = \frac{E}{2(1+\nu)}$$

ν = coefficient de Poisson

5.5.1.3 Etude de la flexion d'une poutre en acier encastrée soumise à une charge constante :

On considère un cas pratique (voir figure 5. 40 ci-dessous) de poutre en acier encastrée soumise à une charge constante p = 1000 N. Le but du cet partie est l'étude de flexion d'une poutre encastrée à l'aide du programme en élément finis et la confrontation des résultats numérique et théorique afin de déterminer le modèle éléments finis adapté à notre exemple. On considère également que :

E=210000 MPa

ν = 0,28

Epaisseur e = 1 mm

Longueur L = 90 mm

Hauteur H = 30 mm

On fait varier deux types de paramètres du maillage :

- le type d'éléments :

- triangles à 3 nœuds ;

- quadrangles à 4 nœuds ;

- la densité des éléments.

Différents maillages (suivant x, suivant y) :

triangle à 3 nœuds	Quadrangle à 4 nœuds
a) maillage 2x4 ; b) maillage 2x8 ; c) maillage 4x12; d) maillage 4x16; e) maillage 6x 20 ; f) maillage 6x 24 ; g) maillage 8x28 ; h) maillage 8x32 ; i) maillage 10x36 ; j) maillage 10x40 ;	a) maillage 2x4 ; b) maillage 2x8 ; c) maillage 4x12; d) maillage 4x16; e) maillage 6x 20 ; f) maillage 6x 24 ; g) maillage 8x28 ; h) maillage 8x32 ; i) maillage 10x36 ; j) maillage 10x40 ;

Table 5. 5: résultats des déplacements d'une poutre en acier encastrée soumise à une charge constante qui utilise l'élément quadrangle à 4 nœuds

Nature des éléments	Maillage	Nb nœuds	Nb éléments	Valeur absolue du déplacement suivant y en mm $\times 10^{-3}$	La solution analytique correspond à $\times 10^{-3}$	Erreur de la solution
Quadrangle à 4 nœuds	a	**15**	8	0440.29	0558.17	21.11%
	b	**27**	16	0510.72	0558.17	8.50%
	c	**65**	48	0538.13	0558.17	3.59%
	d	**85**	64	0544.60	0558.17	2.43%
	e	**147**	120	0550.76	0558.17	1.32%
	f	**175**	144	0552.68	0558.17	0.98%
	g	**261**	224	0555.21	0558.17	0.53%
	h	**297**	256	0556.10	0558.17	0.37%
	i	**407**	360	0557.49	0558.17	0.12%
	j	**451**	400	0558.00	0558.17	0.03%

Table 5. 6: les résultats de déplacements d'une poutre en acier encastrée soumise à une charge constante qui utilise l'élément triangle à 3 nœuds

Nature des éléments	Maillage	Nb nœuds	Nb éléments	Valeur absolue du déplacement suivant y en mm $\times 10^{-3}$	La solution analytique correspond à $\times 10^{-3}$	Erreur de la solution
Triangle à 3 nœuds	a	**15**	16	0264.09	0558.17	52.68%
	b	**27**	32	0332.40	0558.17	40.44%
	c	**65**	96	0460.29	0558.17	17.53%
	d	**85**	128	0472.85	0558.17	15.28%
	e	**147**	240	0512.06	0558.17	8.26%
	f	**175**	288	0516.18	0558.17	7.52%
	g	**261**	448	0532.10	0558.17	4.67%
	h	**297**	512	0533.96	0558.17	4.33%
	i	**407**	720	0541.93	0558.17	2.90%
	j	**451**	800	0542.95	0558.17	2.72%

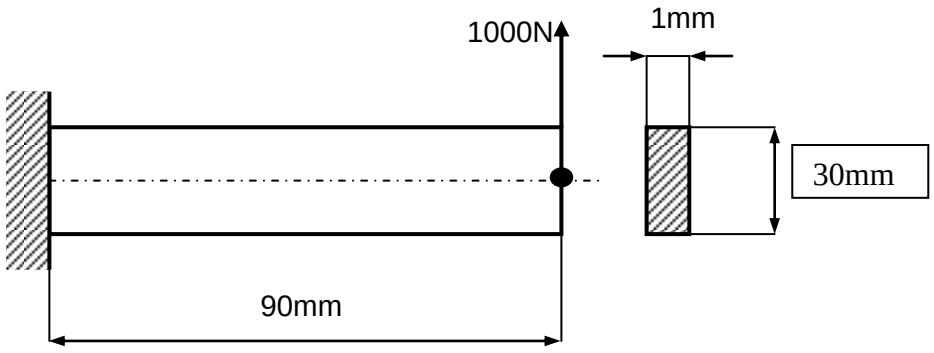


Figure 5. 43 : Poutre encastrée soumise à une charge constante

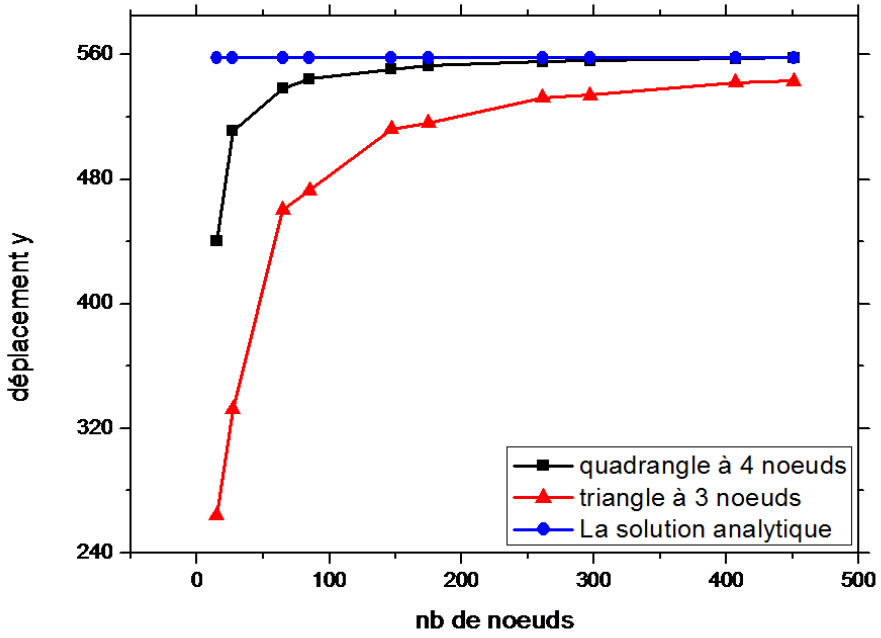


Figure 5. 44 : la variation de déplacement y en fonction de nb de nœuds

La figure 5.44 montre que le triangle à 3 nœuds est très peu performant. Même lorsque le nombre de nœuds augmente, la solution ne tend que très lentement vers la solution exacte.

Ainsi, il ne sert à rien de se servir d'élément triangulaire à 3 nœuds en élevant inconsidérément le nombre de nœuds.

L'élément quadrangle à 4 nœuds est légèrement plus performant que le triangle à 3 nœuds. La solution est rapidement approchée, même avec un faible nombre de nœuds.

On peut donc conclure de cette petite étude que :

- le choix du type d'élément est très important ;

- en flexion, triangle à 3 nœuds et quadrangle à 4 sont des éléments sont économiques mais n'est pas performants.

Comme c'était évident, l'élément triangle à 3 nœuds introduit les plus grandes erreurs par rapport au résultat exact. Les quadrangles à 4 nœuds sont meilleurs que l'élément triangle à 3 nœuds mais pas les meilleurs à être utilisés lors d'une discrétisation.

Solution analytique :

Pour ce problème relativement simple, il existe une solution analytique aisément calculable. D'après les caractéristiques du matériau et les paramètres géométriques fournis par l'énoncé nous pouvons calculer la flèche en utilisant la formule utilisée dans la Résistance des matériaux, on aura:

$$y_{max} = \frac{FL^3}{3EI} + \frac{6FL}{5AG}$$

La flèche maximale calculée grâce à la solution analytique est :

y_{max} =0.55817mm

A = aire de la section droite

$\boldsymbol{I_Z}$= moment quadratique suivant z

E = module de YOUNG

$$G = \frac{E}{2(1+v)}$$

v = coefficient de Poisson

5. 6 Application à des problèmes thermiques :

5. 6.1 Introduction :

Dans cette partie, on présente quelques exemples de transfert thermiques afin de voir la performance de éléments triangulaires à 3 nœuds ou quadrangulaires à 4 nœuds. En effet on a traité problème de conduction test étudiée en l'appliquant à la solution de 2 problèmes dans domaine thermique d'un four a fait l'objet d'un deuxième exemple.

5. 6.1.1 conductions thermiques dans une plaque :

Description du problème :

Dans ce problème, on présente une solution du problème de conduction de chaleur 2D (la Figure 5. 45), à l'aide d'un programme d'élément finis MAT-fem.

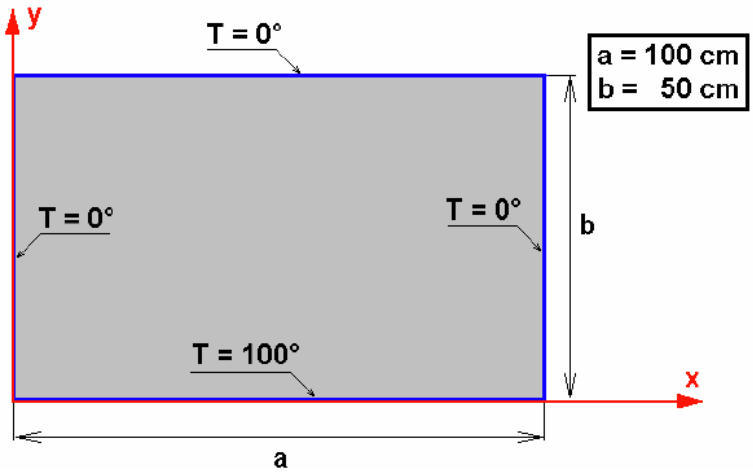


Figure 5. 45 une représentation et dimensionnement en 2 D d'une plaque rectangulaire

Conditions de symétrie :

-Symétrie de la géométrie, on coupe la plaque en 2 verticalement en : x= a/2.

-Symétrie des conditions limites.

Donc : symétrie de la solution

On étudie donc qu'une demi-plaque.

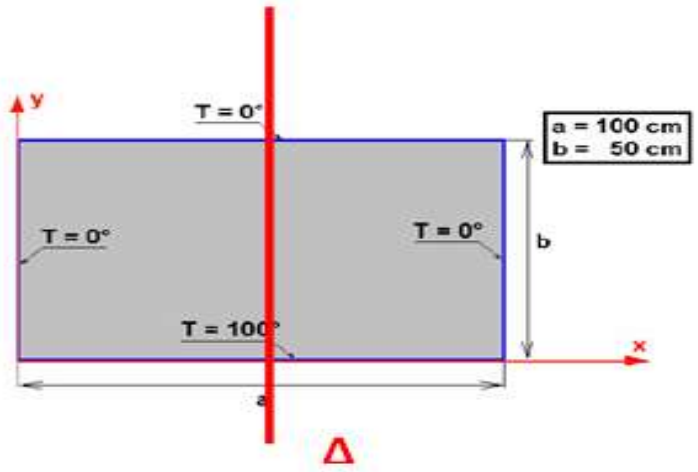


Figure 5. 46 une représentation et dimensionnement en 2 D d'une plaque rectangulaire après la symétrie.

Conditions aux limites:

Conditions de type Dirichlet :

On impose une condition de Dirichlet lorsqu'on fixe une valeur à l'inconnue du problème.

Problème de thermique, l'inconnue : températures sont fixées par les données du problème. On a donc ici affaire à des conditions de Dirichlet.

En effet, on a ici :

T = 0°C, YЄ [0,50], X = 0

T = 0°C, XЄ [0,50], Y = 50

T = 100°C, XЄ [0,50], Y = 0

Conditions de type Neumann :

On impose une condition de Neumann lorsqu'on fixe un chargement. En thermique : on impose un flux(φ).

On a ici :

φ = 0 W, YЄ[0,50], X = 50

Propriétés des matériaux:

Conductivité thermique λ (acier) = 52 W/m².°C ·

Objectif: cherche la distribution de la température nodale la plaque rectangulaire.

On fait varier :

La densité de maillage : **4*4, 5*5, 6*6, 8*8, 10*10, 12*12,**

L'ordre et la topologie de l'élément (**T3** et **Q4**)

-**T** pour topologie Triangle.

-**Q** pour topologie Quadrangle.

-**3 et 4** : Approximations par des fonctions linéaires (ordre 1 : 1 variable).

-NN : nombre de nœud.

-NE : nombre d'élément.

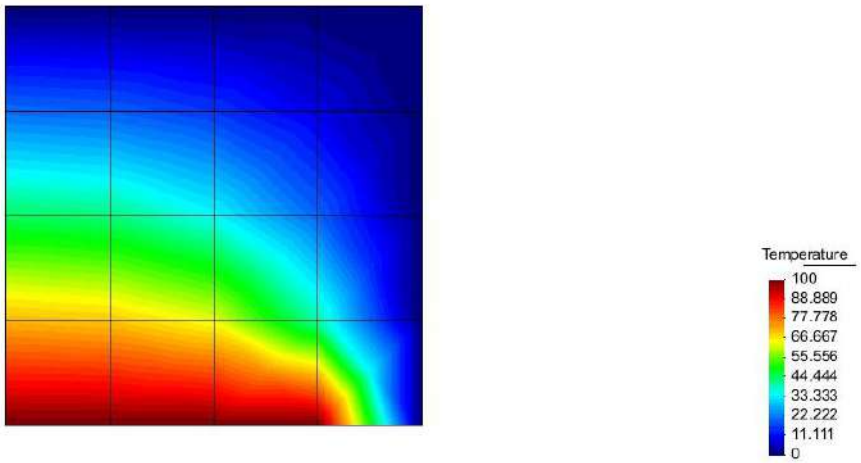


Figure 5. 47 la distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=25 et NE=16

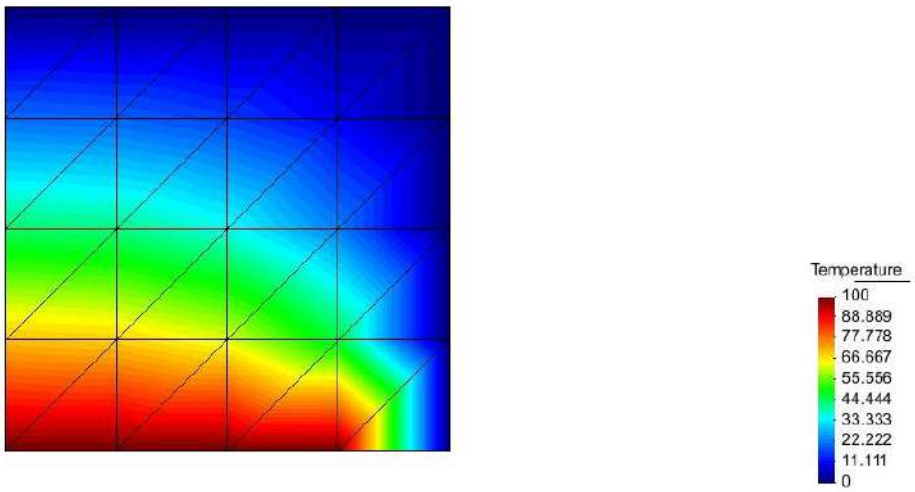


Figure 5. 48 la distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=25 et NE=32

Table 5. 7: représente la température T en °C Pour la distance y varie et x = 0 .

Type d'élément / Y en mètre	l'élément quadrangulaire à 4 nœuds	l'élément triangulaire à 3 nœuds
0,5	0	0
0,375	21,1979	20,962
0,25	44,5667	43,9806
0,125	71,1199	70,4466
0	100	100

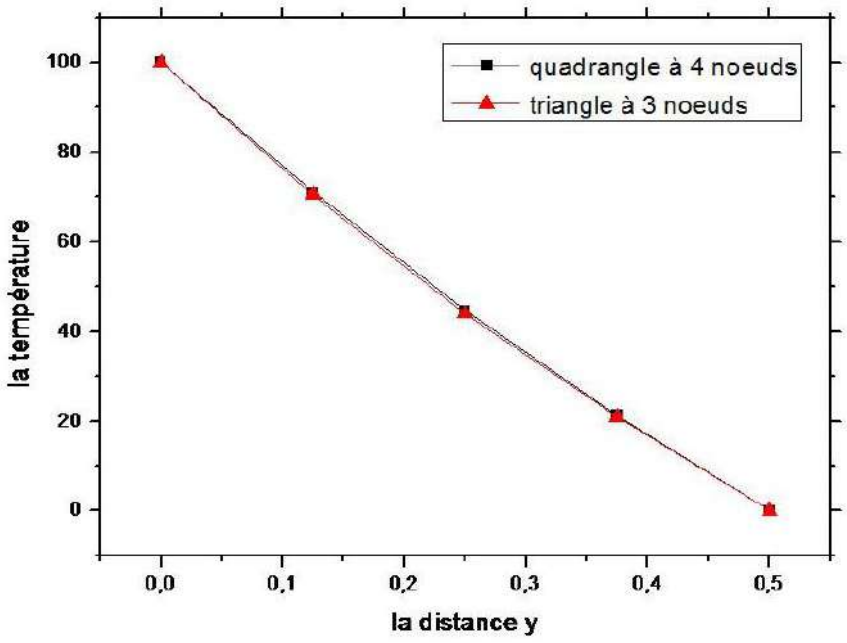


Figure 5. 49 comparaison de 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0

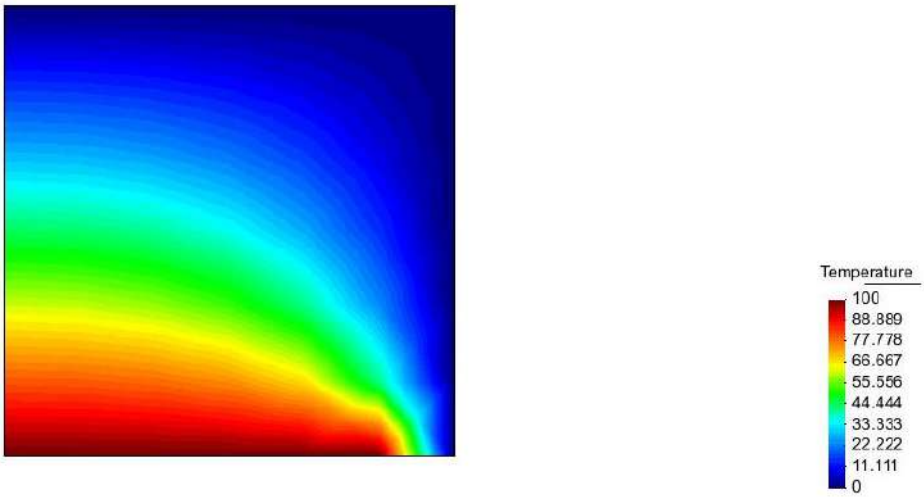


Figure 5. 50 la distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=49 et NE=36

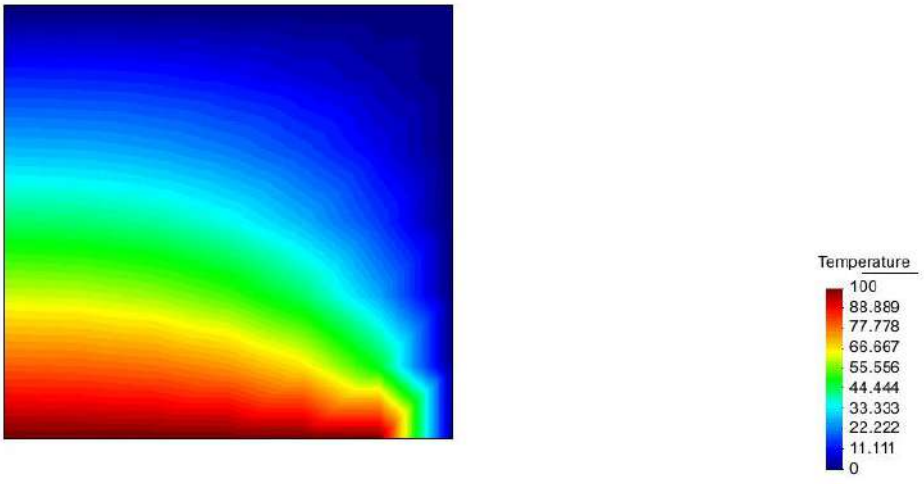


Figure 5. 51 la distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=49 et NE=72

Table 5. 8: représente la température T en °C Pour la distance y varie et x = 0 .

Type d'élément / Y en mètre	l'élément quadrangulaire à 4 nœuds	l'élément triangulaire à 3 nœuds
0,5	0	0
0,41667	14,0022	13,9358
0,33333	28,6667	28,5078
0,25	44,529	44,262
0,16667	61,8566	61,5333
0,08333	80,5255	80,2892
0	100	100

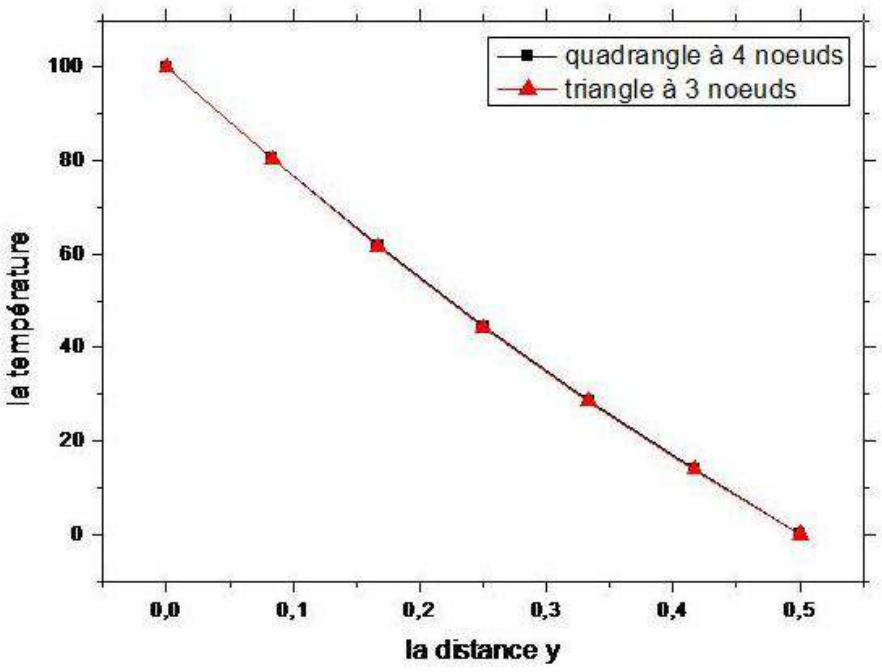


Figure 5. 52 comparaison de 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0

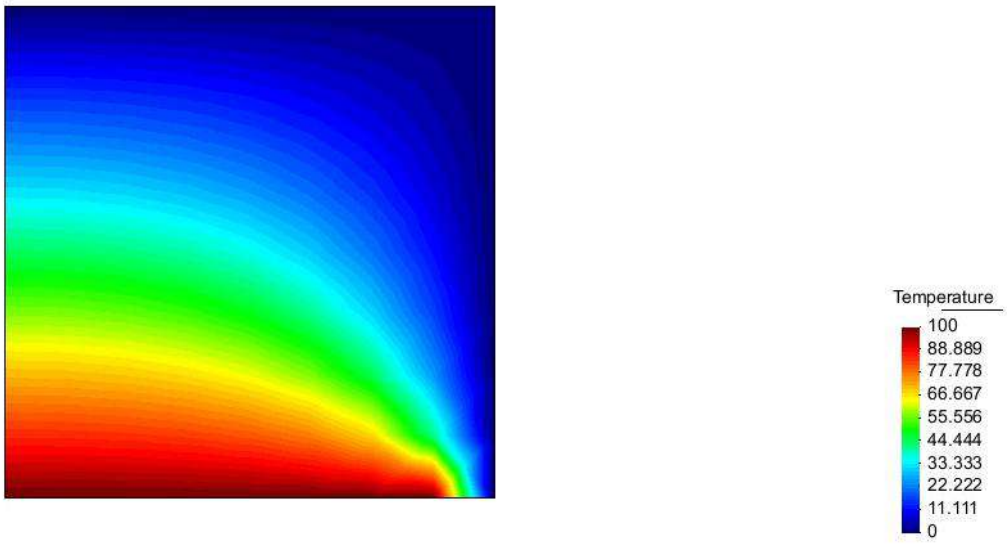


Figure 5. 53 la distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=81 et NE=64

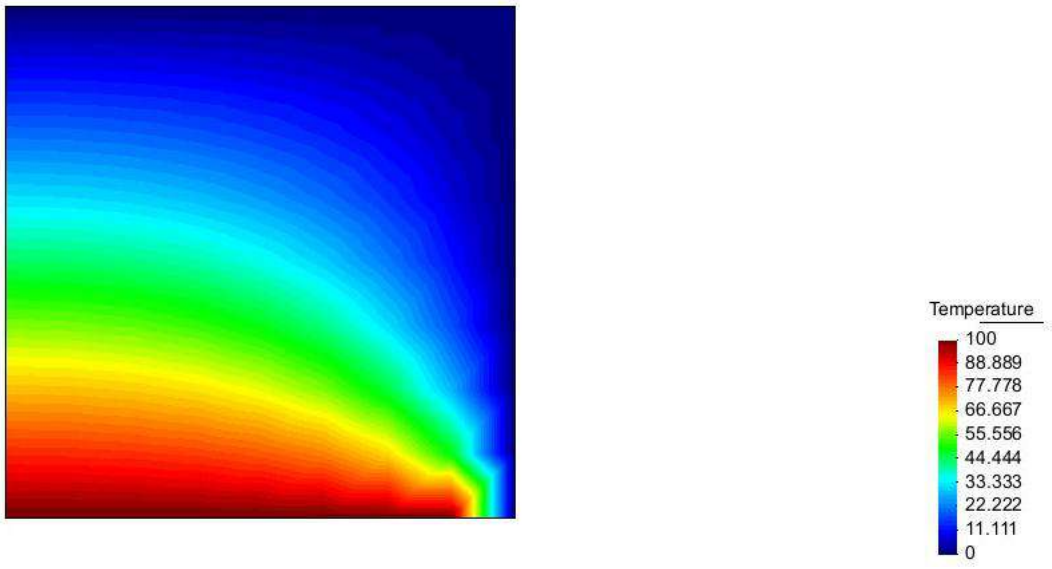


Figure 5. 54 la distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=81 et NE=128

Table 5. 9 : représente la température T en °C Pour la distance y varie et x = 0 .

Type d'élément / Y en mètre	l'élément quadrangulaire à 4 nœuds	l'élément triangulaire à 3 nœuds
0,5	0	0
0,4375	10,4709	10,443
0,375	21,2217	21,1592
0,3125	32,5061	32,3999
0,25	44,5199	44,3683
0,1875	57,3666	57,1859
0,125	71,0245	70,8548
0,0625	85,3313	85,2264
0	100	100

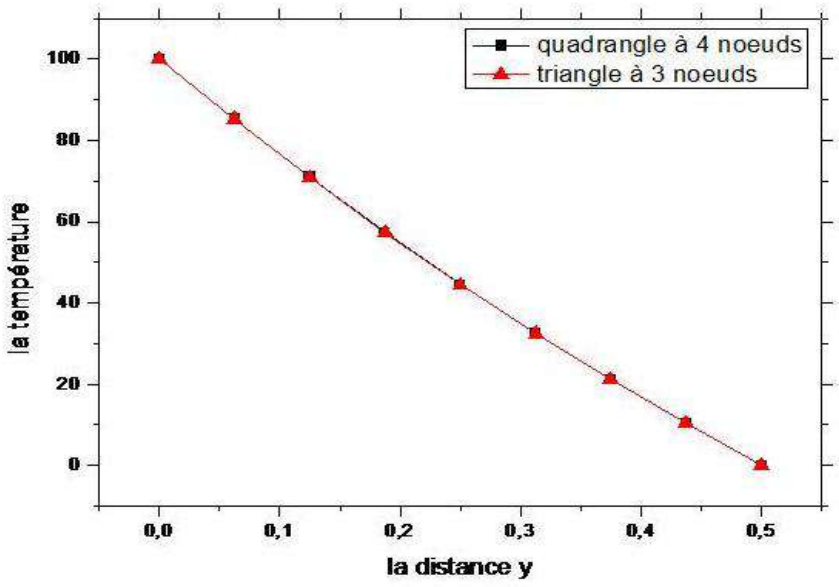


Figure 5. 55 comparaison de 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0

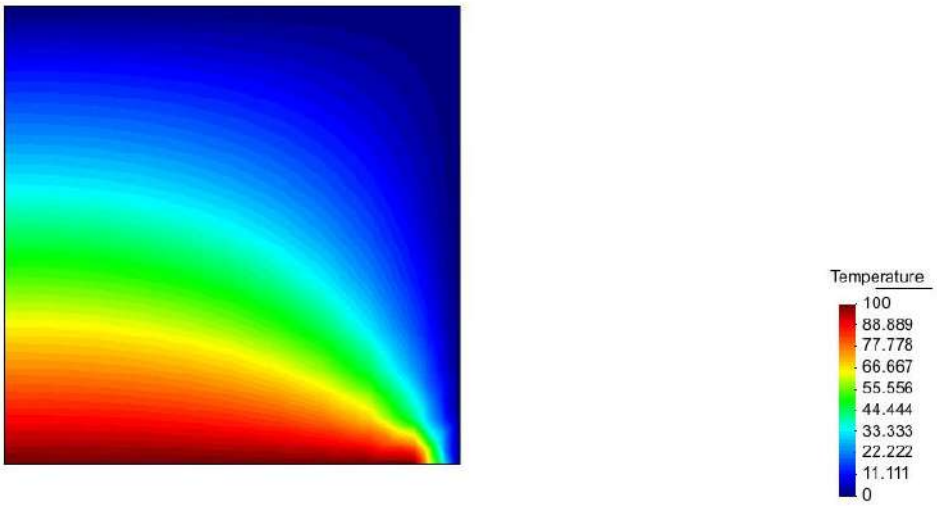


Figure 5. 56 la distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=121 et NE=100

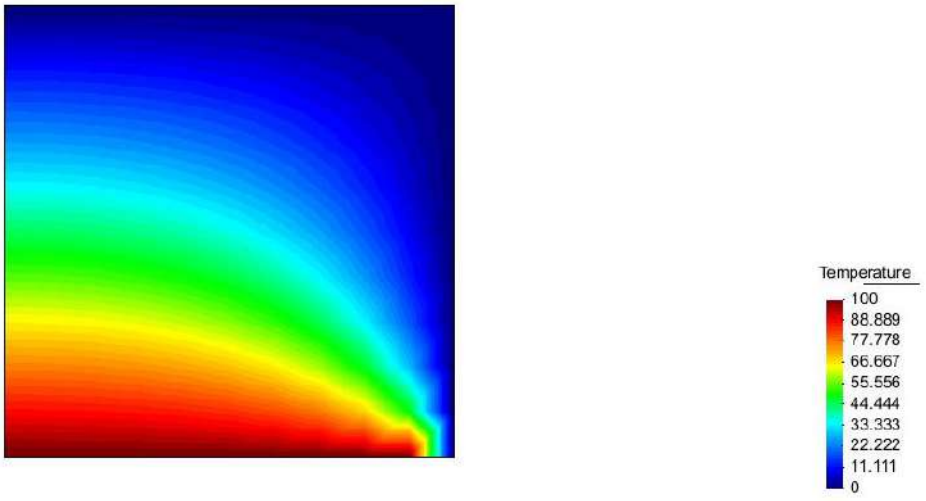


Figure 5. 57 la distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=121 et NE=200

Table 5. 10 : représente la température T en °C Pour la distance y varie et x = 0 .

Type d'élément / Y en mètre	l'élément quadrangulaire à 4 nœuds	l'élément triangulaire à 3 nœuds
0,5	0	0
0,45	8,36558	8,35135
0,4	16,8746	16,8438
0,35	25,6623	25,6112
0,3	34,8464	34,7719
0,25	44,5165	44,419
0,2	54,7219	54,6081
0,15	65,4608	65,3453
0,1	76,673	76,577
0,05	88,241	88,186
0	100	100

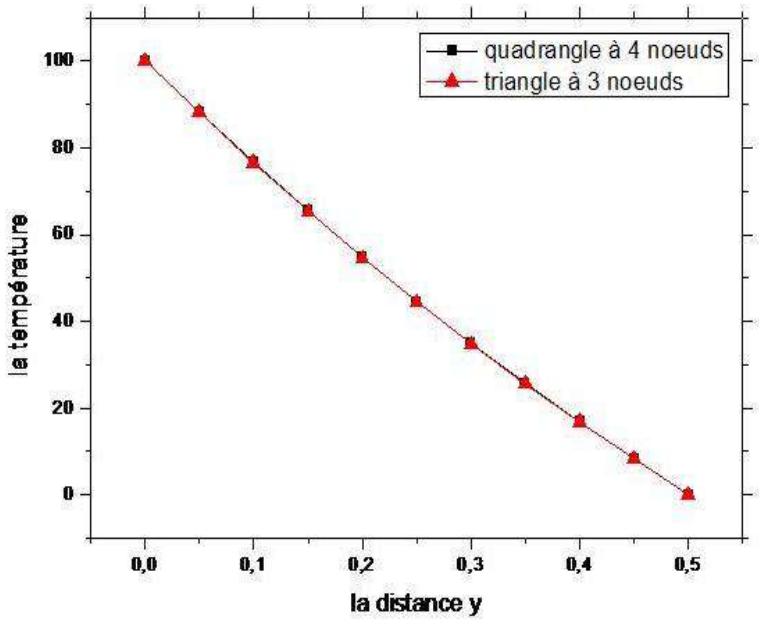


Figure 5. 58 comparaison de 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0

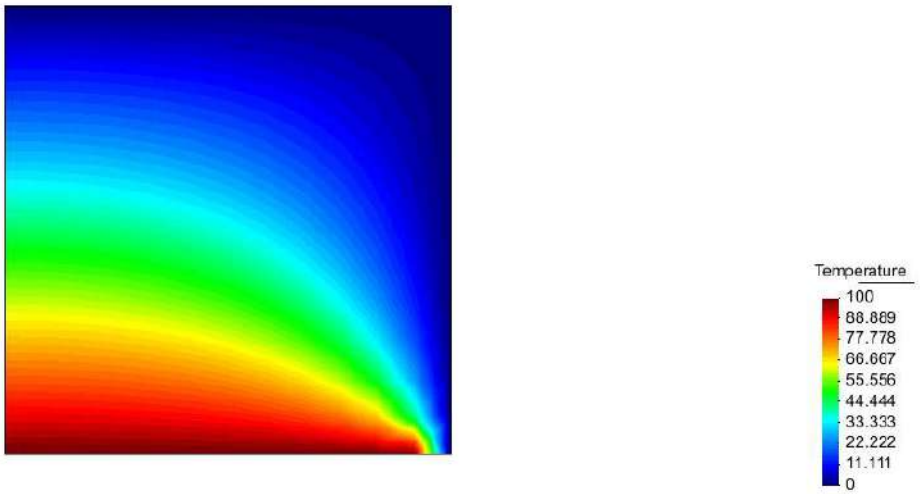


Figure 5. 59 la distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=169 et NE=144

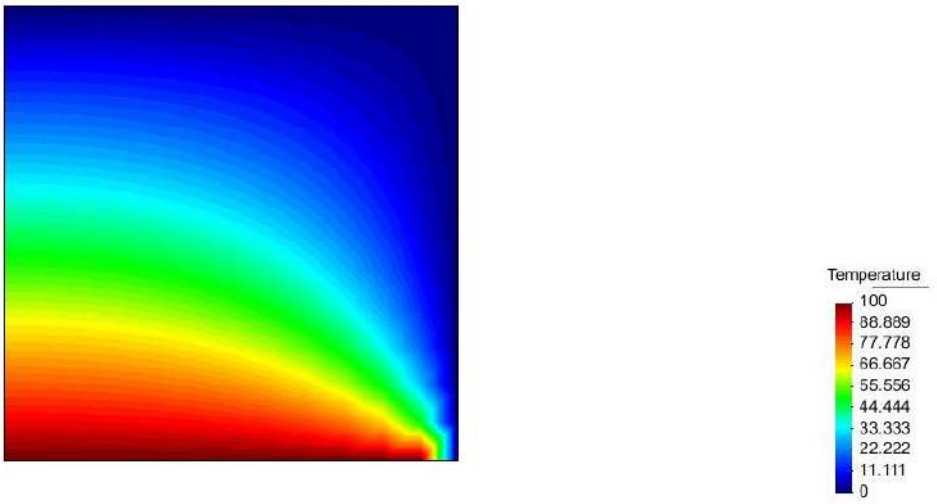


Figure 5. 60 la distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=169 et NE=288

Type d'élément / Y en mètre	l'élément quadrangulaire à 4 nœuds	l'élément triangulaire à 3 nœuds
0,5	0	0
0,45833	6,96636	6,95813
0,41667	14,0157	13,9983
0,375	21,2279	21,1997
0,33333	28,6761	28,6352
0,29167	36,4227	36,368
0,25	44,5148	44,4469
0,20833	52,9795	52,9017
0,16667	61,8191	61,7378
0,125	71,0081	70,9327
0,08333	80,4919	80,4333
0,04167	90,1897	90,1575
0	100	100

Table 5. 11 : représente la température T en °C Pour la distance y varie et x = 0 .

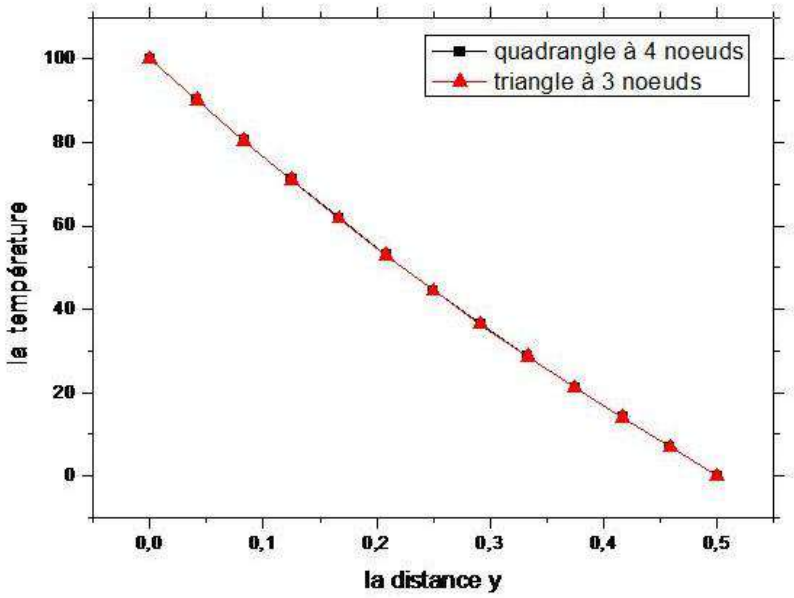


Figure 5. 61 comparaison de 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0.

5. 6.1.2 Modélisations thermiques de four :

Description du problème: ·

Nous modélisons le transfert thermique d'un four 2D. En utilisant la symétrie, nous pouvons limiter le champ d'application de la modélisation à une seule section du four, comme le montre l'illustration (Figure 5. 63). Cette mise en œuvre de symétrie simplifie grandement l'effort requis pour l'analyse thermique.



Figure 5. 62 une représentation d'un four réel [22]

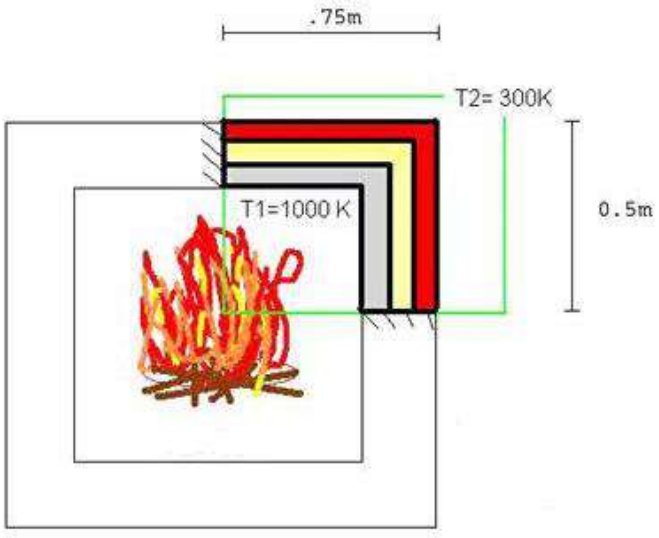


Figure 5. 63 une représentation et dimensionnement en 2 D d'un four après la symétrie.

Conditions aux limites:

Conditions de type Dirichlet :

On impose une condition de Dirichlet lorsqu'on fixe une valeur à l'inconnue du problème.

Problème de thermique, l'inconnue : températures sont fixées par les données du problème. On a donc ici affaire à des conditions de Dirichlet.

1) Les lignes de symétrie de la pièce d'angle peuvent être supposées être isolé.

2) La limite la plus extérieure a une température constante de 300 K=27 °C.

3) La limite plus à l'intérieur a une température constante de 1000 K=727 °C.

Propriétés des matériaux:

Conductivité λ (Brique terre crue pleine) = 1,1W/m².°C ·

Dimensions :
Longueur = 1,5 m

Largeur = 1 m

Objectif: cherche la distribution de la température nodale et le flux de chaleur du four.

On fait varier :

La densité de maillage : **4*4, 5*5, 6*6, 8*8, 10*10, 12*12,**

L'ordre et la topologie de l'élément (**T3** et **Q4**)

-**T** pour topologie Triangle

-**Q** pour topologie Quadrangle

-**3 et 4** : Approximations par des fonctions linéaires (ordre 1 : 1 variable)

-**NN** : nombre de nœud, **NE** : nombre d'élément

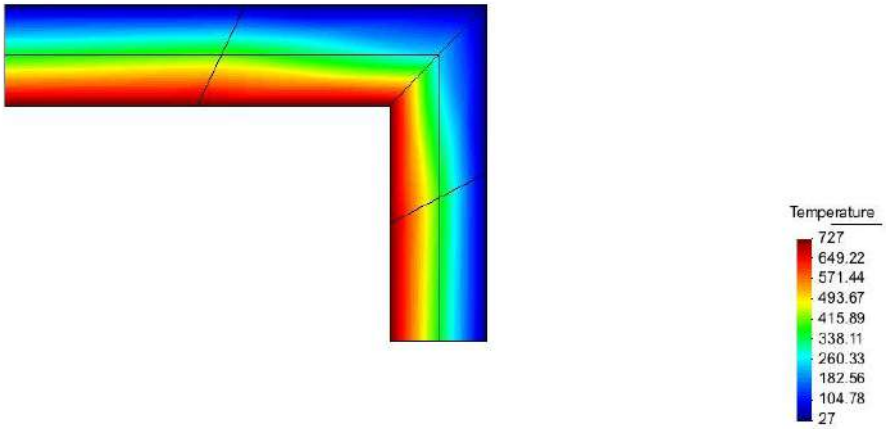


Figure 5. 64 la distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=15 et NE=8

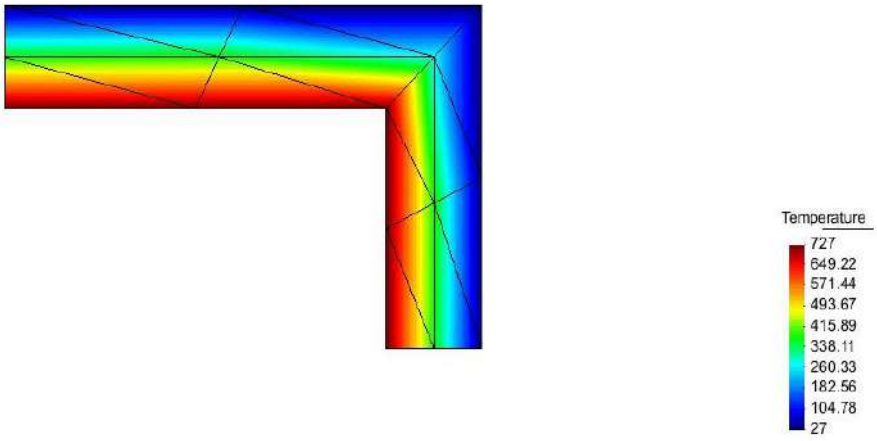


Figure 5. 65 la distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=15et NE=16

Type d'élément / Y en mètre	l'élément quadrangulaire à 4 nœuds	l'élément triangulaire à 3 nœuds
0,5	27	27
0,425	364,483	377
0,35	727	727

Table 5. 12 : représente la température T en °C Pour la distance y varie et x = 0.

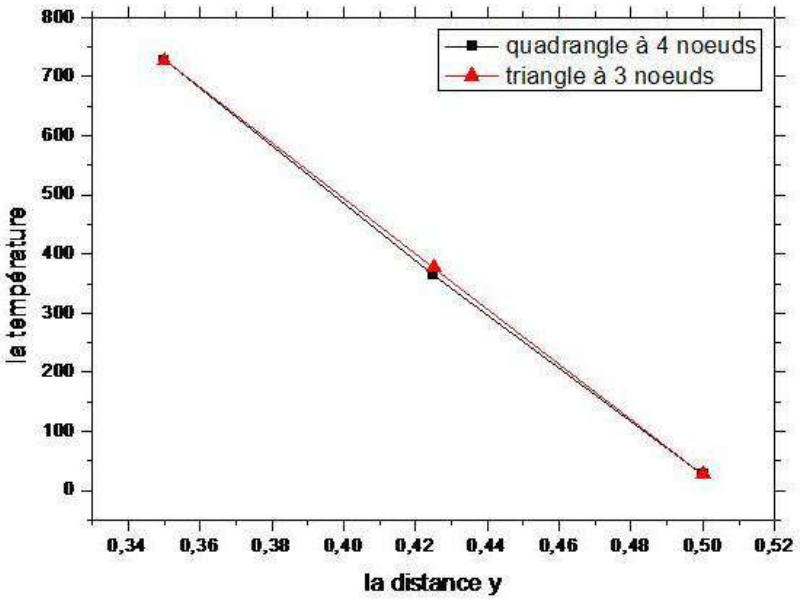


Figure 5. 66 comparaison entre 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0

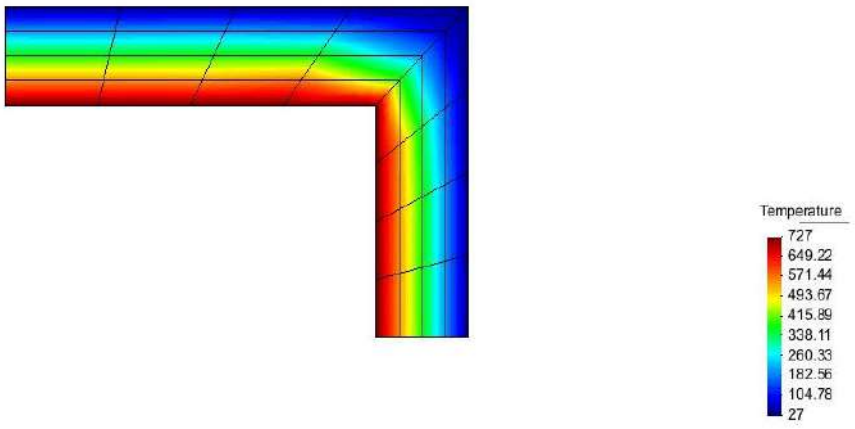


Figure 5. 67 la distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=45 et NE=32

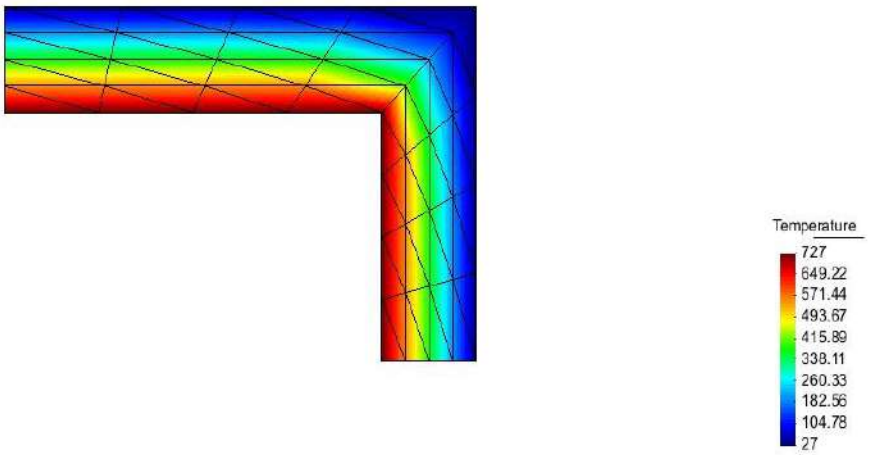


Figure 5. 68 la distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=45 et NE=64

Type d'élément / Y en mètre	l'élément quadrangulaire à 4 nœuds	l'élément triangulaire à 3 nœuds
0,5	27	27
0,4625	202,23199	202,00301
0,425	377,30301	377,00299
0,3875	551,73999	552,00201
0,35	727	727

Table 5. 13 : représente la température T en °C Pour la distance y varie et x = 0 .

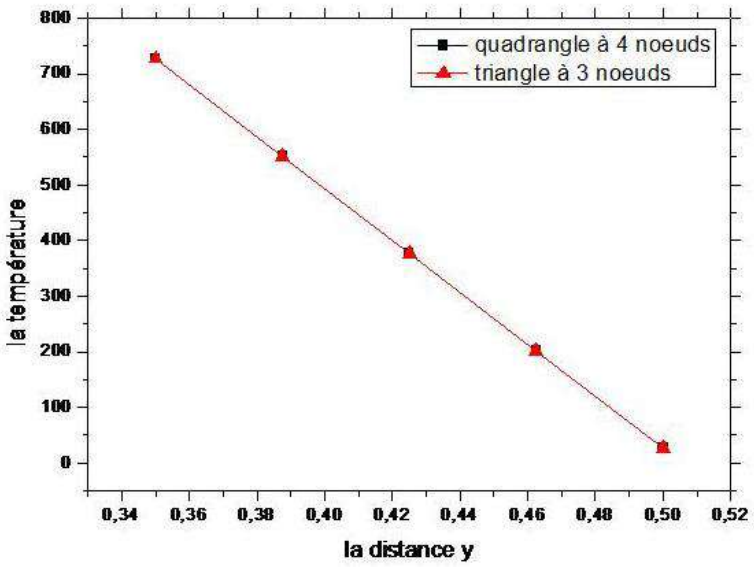


Figure 5. 69 comparaison entre 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0

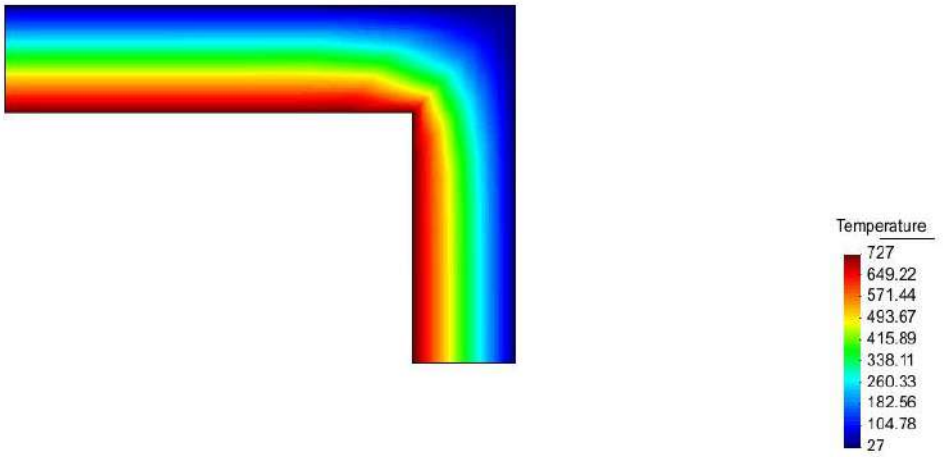


Figure 5. 70 la distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=91 et NE=72

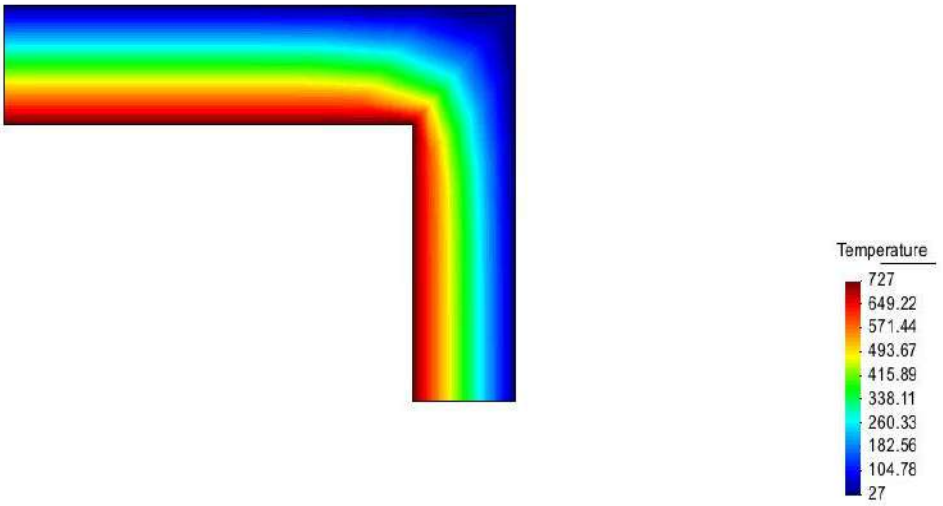


Figure 5. 71 la distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=91 et NE=144

Type d'élément / Y en mètre	l'élément quadrangulaire à 4 nœuds	l'élément triangulaire à 3 nœuds
0,5	27	27
0,475	143,66299	143,66701
0,45	260,32501	260,33401
0,425	376,99799	377
0,4	493,681	493,66699
0,375	610,32703	610,33301
0,35	727	727

Table 5. 14 : représente la température T en °C Pour la distance y varie et x = 0 .

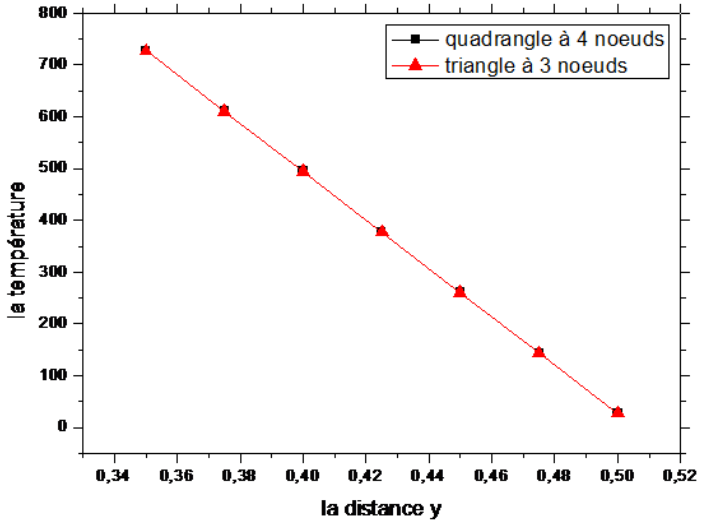


Figure 5. 72 comparaison entre 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0

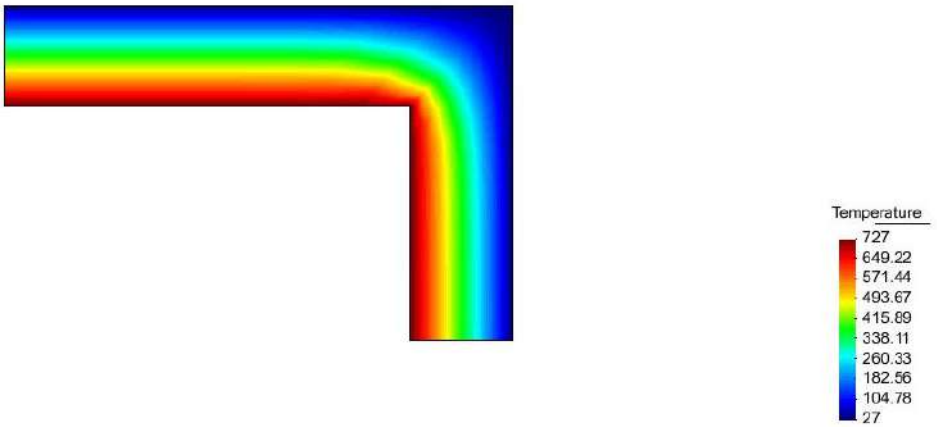


Figure 5. 73 la distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=153 et NE=128

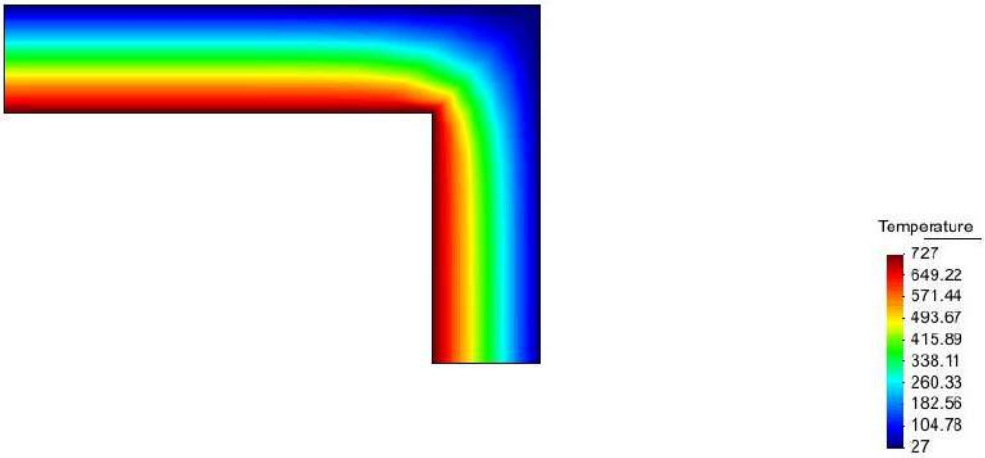


Figure 5. 74 la distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=153 et NE=256

Type d'élément / Y en mètre	l'élément quadrangulaire à 4 nœuds	l'élément triangulaire à 3 nœuds
0,5	27	27
0,48125	114,5	114,5
0,4625	202	202
0,44375	289,5	289,5
0,425	377	377
0,40625	464,5	464,5
0,3875	552,00098	552
0,36875	639,5	639,5
0,35	727	727

Table 5. 15 : représente la température T en °C Pour la distance y varie et x = 0 .

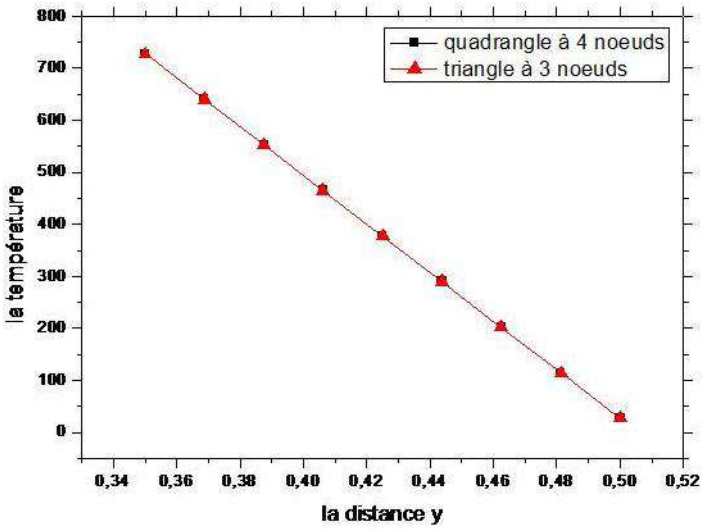


Figure 5. 75 comparaison entre 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0

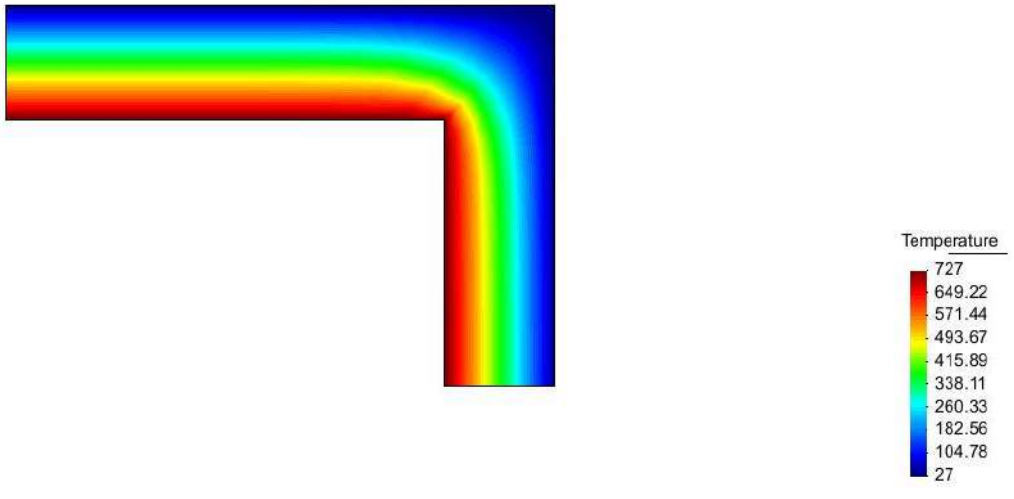


Figure 5. 76 la distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=231 et NE=200

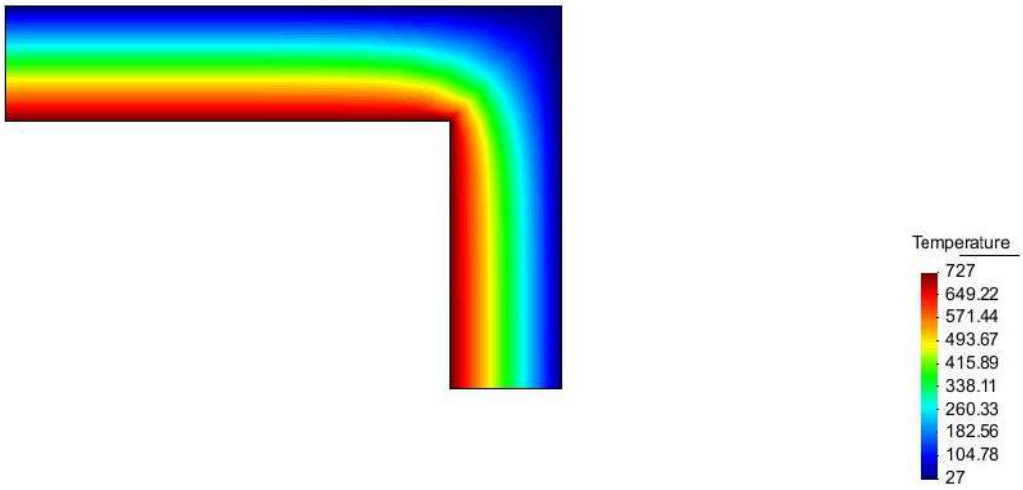


Figure 5. 77 la distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=231 et NE=400

Type d'élément / Y en mètre	l'élément quadrangulaire à 4 nœuds	l'élément triangulaire à 3 nœuds
0,5	27	27
0,485	97	96,9998
0,47	167	167
0,455	237	237
0,44	307	307
0,425	377	377
0,41	447	447
0,395	517	517
0,38	587	587
0,365	657	657
0,35	727	727

Table 5. 16 : représente la température T en °C Pour la distance y varie et x = 0 .

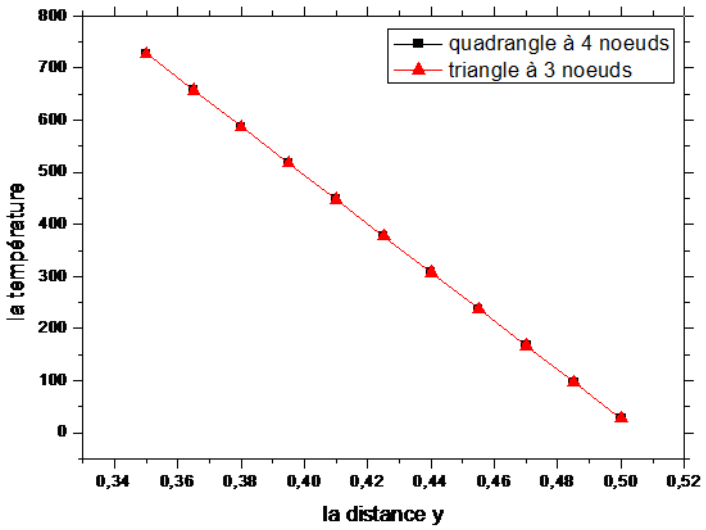


Figure 5. 78 comparaison entre 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0

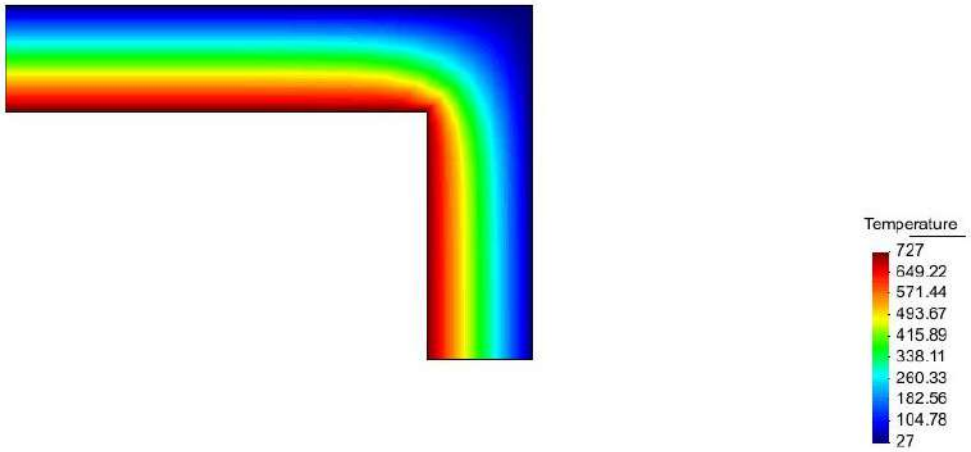


Figure 5. 79 la distribution de la température nodale en utilise l'élément quadrangulaires à 4 nœuds avec NN=325 et NE=288

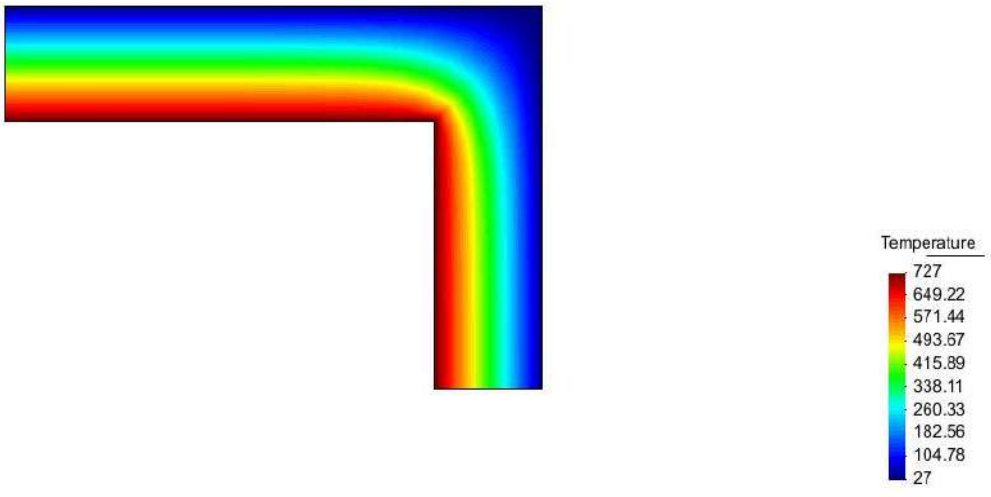


Figure 5. 80 la distribution de la température nodale en utilise l'élément triangulaires à 3 nœuds avec NN=325 et NE=288

Type d'élément / Y en mètre	l'élément quadrangulaire à 4 nœuds	l'élément triangulaire à 3 nœuds
0,5	27	27
0,4875	85,3333	85,3332
0,475	143,66701	143,666
0,4625	202	202
0,45	260,33301	260,33301
0,4375	318,66599	318,66599
0,425	377	376,99899
0,4125	435,33301	435,33301
0,4	493,66699	493,66599
0,3875	552	552
0,375	610,33301	610,33301
0,3625	668,66699	668,66699
0,35	727	727

Table 5. 17 : représente la température T en °C Pour la distance y varie et x = 0 .

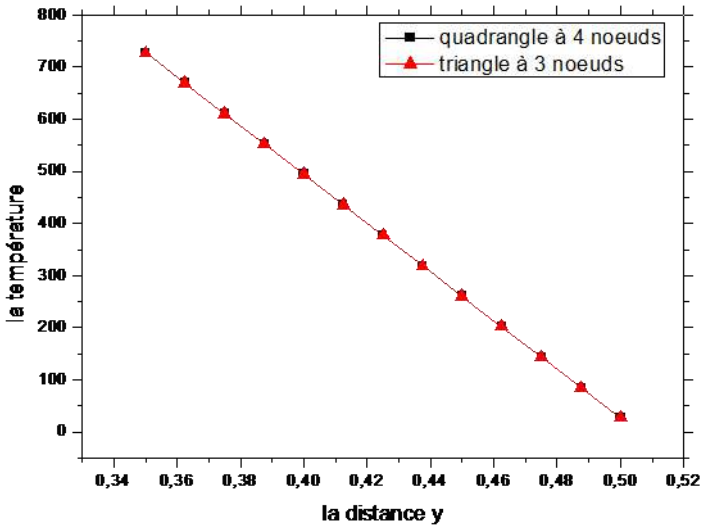


Figure 5. 81 comparaison entre 2 élément (triangle et quadrangle) la variation de température en fonction de la distance y et x=0

L’analyse de ces résultats obtenus nous permet de conclure que :

- Plus la densité de maillage est grande, plus on observe de grandes variations dans la distribution de température (plus le maillage est raffiné, plus la température est constante dans la plaque et le four).
- Les figures montrent que le triangle à 3 nœuds est très peu performant. En effet, lorsque le nombre de nœuds augmente la solution convergent très lentement vers la même solution pour les 2 éléments (triangle à 3 nœuds et quadrangle à 4 nœud).
- L'élément quadrangle à 4 nœuds est légèrement plus performant que le triangle à 3 nœuds. La solution est rapidement approchée, même avec un faible nombre de nœuds.

Conclusion générale:

Dans cette étude, nous avons fixé comme objectif le développement d'un pré-processeur en FORTRAN 77 afin de répondre au besoin d'un générateur automatique de maillage.

Dans le présent travail on a reprise les routines des travaux [3,14,15,17] pour développer un code de structure modulaire .des modifications sont introduits afin de donner au code développé une structure cohérente .une vérifications des résultats obtenus par [3,15] seront vérifiés.

La technique utilisée dans le mailleur développé MESH_2D se base sur la méthode de génération de maillage par blocs. On a adopté une stratégie qui consiste à éliminer la répétition des nœuds, ce qui permet aux éléments de différents types (linéaires ou quadratiques) d'être générés simultanément lors de la génération de maillage. Après que les éléments de types souhaités ont été générés dans les différentes régions, les nœuds sont réorganisés et le maillage est terminé.

Par ailleurs, on a essayé de minimiser la quantité de données d'entrée nécessaire pour décrire un problème par éléments finis lors de la conception du fichier de données du code ; ce qui facilite la manipulation de ce pré-processeur.

D'autre part, deux caractéristiques essentielles ont été adoptées lors de l'élaboration du code développé afin d'améliorer la procédure de génération et donc de minimiser le temps pris pour l'utilisateur de générer un maillage :

- La génération du maillage se fait autour d'un axe de symétrie perpendiculaire à chaque génération d'éléments de forme rectangulaire ou triangulaire.
- La structure modulaire adoptée lors de l'élaboration de ce code générateur de maillage lui donne plus d'efficacité et de souplesse sur les plans de développement ou de contrôle. Elle permet aux utilisateurs du mailleur de remplacer ou améliorer facilement les sous programmes ou bien d'effacer les sous programmes qui ne sont pas nécessaires pour leur application, par exemple, la renumérotation des nœuds.

Cependant, l'inconvénient principal du maillage par la méthode des blocs, parfois connues sous le nom de méthode de superéléments, est que la phase de décomposition de l'objet en sous-domaines à 3 ou 4 côtés doit être faite par l'utilisateur. Cette décomposition dépend de la nature du problème et de la connaissance des éléments finis de l'utilisateur. Donc pour un mailleur entièrement automatique ceci n'est pas acceptable.

A la fin de ce mémoire, on a voulu procéder à l'application des techniques développées précédemment à des problèmes physiques réels (solide et thermique).

Pour cela, on a adapté les fichiers des résultats de notre code mailleur aux fichiers d'entrée de deux codes solveurs existants utilisant l'analyse par la méthode des éléments finis. En effet, on a présenté une analyse des contraintes pour quelques exemples en 2D afin de valider et tester le programme MESH_2D pour des problèmes dans le domaine thermique et solide.

D'après les résultats obtenus, on a montré que notre mailleur s'avère efficace et répond à l'objectif assigné puisqu'il comporte les caractéristiques suivantes :

1. Il est entièrement automatique et facilement contrôlable (minimiser l'intervention de l'usager) ;
2. Il peut produire un maillage avec des éléments finis bien conditionnés et fiables.
3. Le maillage généré vérifie la condition de fidélité à la géométrie du problème.
4. Il convient également de contrôler la forme des éléments finis générés.
5. Il est capable de raffiner dans les zones où les gradients de solution sont importants.

Recommandations pour les travaux futurs :

En fait, une méthode de maillage robuste combinera toutes les méthodes déjà mentionnées dans le chapitre 2 et essaiera de tirer profit des divers avantages propres à chacune des méthodes après avoir décomposé le domaine en plusieurs blocs.

Comme perspectives, ce code mailleur développé au cours de ce travail, on propose d'apporter les améliorations suivantes :

- Une interface interactive peut rendre le code développé plus pratique et facile à utiliser.
- On doit adapter le OUTPUT de notre code au INPUT de quelques codes commerciaux souvent utilisés la communauté scientifique comme ANSYS, FLUENT et ABAQUS.
- Une version 3D du mailleur s'avère indispensable à l'état actuel

Références bibliographiques :

[1] COUPEZ, T. (1991). Grandes transformations et remaillage automatique. Thèse de doctorat, École nationale supérieure des mines de Paris.

[2] R. Löhner and P. Parikh ,(1988), Generation of three-dimensional unstructured grids by the advancing-front method, Int. J. Num. Meth. Fluids 8, 1135-1149.

[3] L. L. Durocher and A. Gasper, (1979). A versatile two- dimensional mesh generator with automatic bandwidth reduction. Comput. Struct. 10, 561-575

[4] J.F. Thompson, Z.U.A. Warsi and C.W. Mastin (1985), Numerical grids generation, foundations and applications, North Holland.

[5] D. Euvrard , (1994). Résolution des équations aux dérivées partielles de la physique, de la mécanique et des sciences de l'ingénieur. Différences finies, éléments finis, problèmes en domaines non bornés, 3 éd., Masson.

[6] R.W. Lewis, Y. Zheng and D.T. Gethin ,(1996), Three- dimensional unstructured mesh generation: Part 3. Volume meshes, Comput. Methods Appl. Mech. Engrg. 134, 285-310.

[7] MOULARDL, (1994). Optimisation de Maillages Non-Structurés: Application à la Génération, a la correction et a l'adaptation. Thèse de doctorat, Université Joseph Fourier-Grenoble.

[8] O. Hassan, N.P. Weatherill, J. Peraire, K. Morgan, E.J. Probert (1995), Generation and adaption of unstructured meshes, COSMASE, Advanced course on grid generation.

[9] A. Rassineux (1995), Maillage automatique tridimensionnel par une méthode frontale pour la méthode des éléments finis, Thèse de Doctorat de l'Université Henri Poincaré - Nancy I.

[10] P.L. George. (1990), Génération automatique de maillages. Applications aux méthodes d'éléments finis, Paris : Masson.

[11] K. Ho-Le, (1988). Finite Element Mesh Generation Methods: A Review and Classification. Computer Aided Design, vol. 20, No. 1, pp. 27-38.

[12] Taniguchi Takeo , (1988). An interactive automatic mesh generator for the microcomputer. Computers *and* Structures, vol. 30, No. 3, pp. 715-722.

[13] Kai-Lung Lin, Heiu-Jou Shaw, (1991). Two-Dimensional Orthogonal Grid Generation Techniques, Computers *and* Structures, vol. 41, No. 4, pp. 569-583 .

[14] V.N. Kaliakin, (1992). A Simple Coordinate Determination Scheme for Two-Dimensional Mesh Generation, *Computers and Structures*, vol. 43, No. 3, pp. 505-516 .

[15] **P.T.S. Ho**, (1992). MSHGEN - a FORTRAN program on automatic two-dimensional and three-dimensional triangular mesh generation, Advances in Engineering Software, vol. 14, No. 1, pp. 61-75.

[16] Yuan-Chong Zhang, Jia-Ju Yin, Lei Luo, (1993). A two-dimensional mesh generator with boundary conditions generation, *Computers and Structures*, vol. 46, pp. 175-181.

[17] **C.K. Lee, S.H. Lo**, (1994). A new scheme for the generation of a graded quadrilateral mesh, Computers and Structures, vol. 52, No. 5, pp. 847-857.

[18] H. Borouchaki, George, P. L, F. Hecht, P. Laug and E. Saltel, (1995). Delaunay mesh generation governed by metric specifications. Part 1: Algorithms, *Finite Elements in Analysis and Design*, Vol. 25, pp. 61-83.

[19] M. H. Kadivar, H. Sharifi, (1996). Double mapping of isoparametric mesh generation, Computers *and* Structures, vol. 59, No. 3, pp. 471-477 .

[20] Borouchaki Houman , Frey Pascal J ,George Paul Louis , (1997). Maillage de surfaces Parametric surfaces meshing. Part III: Quadrilateral elements.Comptes Rendus de l'Académie des Sciences , Vol. 325, pp. 551-556

[21] E. Oñate, 2009.Structural Analysis with the Finite Element Method. Linear Statics. Volume 1. Basis and Solids.

[22] J. N. Reddy , 2006.Introduction to the Finite Element Method , Third Edition, McGraw-Hill, New York.

[23] R. Löhner ,(1988), Some useful Data Structures for the Generation of unstructured grids, Commun. numer. methods eng. 4, 123-135.

[24] F. Ghassemi, (1982).Automatic mesh generation scheme for two- and three-dimensional curved surfaces. Comput. Struct. 15, 613-622

[25] A. Rassineux (1997), Maillage automatique tridimensionnel par une technique frontale et respect d'une carte de taille, Revue européenne des éléments finis 6(1), 43-70.

[26] M.S. Shephard and M.K. Georges (1991), Automatic three-dimensional mesh generation technique by the finite octree technique, Int. j. numer. methods eng. 32, 709-749.

[27] J. Carnet (1978), Une méthode heuristique de maillage dans le plan pour la mise en œuvre des éléments finis, Thèse, Paris.

[28] Aboubacar sidiki Sambou Bagayogo, (2001).Géneration de maillage de peau pour géométries complexes par decomposition spatiale hierarchique. Thèse de doctorat, Ecole Polytechnique (Montreal, Canada).

[29] GEORGE, P et BOROUCHAKHI, (1997). Triangulation Delaunay et maillage, application aux éléments finis . Editions HERMES, Paris.

[30] GOLGOLABA, (1989). Mailleurs 3d automatiques pour géométries complexes. Rapport technique, INRIA.

[31] D. Zimmerle, (1986).Generation of variable order elements by superelement generation. Comput. Struct. 22, 291-297

[32] P.L. George (1993), Construction et Modification de Maillages,Guide Modulef 3.

[33] P.L. George, (1996). Automatic Mesh Generation and Finite Element Computation, in Handbook of Numerical Analysis. Amsterdam, 4, pp. 69-190.

[34] P.J. Frey and P.L. George, (2000). Mesh Generation: Application to Finite Elements. Hermes Science Publishing, Oxford.

[35] H. Samet (1990), The Design and Analysis of Spatial Data Structures, Addison-Wesley Publishing Co.

[36] F. Hermeline (1980), Une méthode automatique de maillage en dimension n, Thèse, Université Paris VI, Paris.

[37] E.-R. Wu, (1980). Some findings in using the program, 'A versatile two-dimensional mesh generator with automatic bandwidth reduction.' Comput. Struct. 12, 181-183

[38] E.-R. Wu, (1982).Techniques to avoid duplicate nodes and relax restrictions on the superelement numbering in a mesh generator. Comput. Struct. 15, 419-422

[39] O. C. Zienkiewicz and D. V. Philips, (1971). An automatic mesh generation scheme for plane and curved surfaces by isoparametric co-ordinates. Mt. J. Numer. Meth. Engng 3, 519-528

[40] H. Borouchaki, P.L. George and S.H. Lo (1996), Optimal Delaunay point insertion, Int. j. numer. methods eng. 39(20), 3407-3438.

[41] J. Frykestig (1994), Advancing front mesh generation techniques with application to the finite element method, Ph.D. Thesis, Chalmers Tekniska Högskola, Sweden.

[42] M.A. Yerry and M.S. Shephard (1984), Automatic three-dimensional mesh generation by the modified-octree technique, Int. j. numer. methods eng. 20, 1965-1990.

[43] J.D. Muller, P.L. Roe and H. Deconinck (1992), A frontal approach for node generation in Delaunay triangulations. Unstructured grid methods for advection dominated flows., VKI Lecture notes, pp. 91-97, AGARD Publication R-787.

[44] J.-P. Pelle, P. Beckers et L. Gallimard , (1996). Estimations des erreurs de discrétisation et analyses adaptatives. Application à l'automatisation des calculs éléments finis, Cours IPSI,

[45] M. Berger (1978), Géométrie tome 3: convexes et polytopes, polyèdres réguliers, aires et volumes, Fernand Nathan, Paris.

[46] D.J. Mavriplis (1992), An advancing front Delaunay triangulation algorithm designed for robustness, ICASE report 92-49.

[47] P. L. George, (1991). Génération de maillage par une méthode de type voronoi , Rapport de Recherche 1398, INRIA.

[48] P.J. Frey, H. Borouchaki and P.L. George (1996), Delaunay Tetrahedralization using an Advancing-Front Approach, in 5th International Meshing Roundtable, Pittsburg, PA, 31-43.

[49] P.J. Frey, H. Borouchaki and P.L. George (1997), 3D Delaunay mesh generation coupled with an advancing-front approach, to appear in Comp. Meth. in Appl. Mech. and Engng..

[50] F. Hermeline. (1982), Triangulation automatique d'un polyèdre de dimension n , RAIRO, Analyse numérique 16, no. 3, p. 211,242.

[51] J. Ruppert and R. Seidel (1992), On the difficulty of triangulating three-dimensional nonconvex polyhedra, Discrete Comput. Geom. 7, 227-253.

[52] M.L. Merriam (1991), An efficient advancing front algorithm for Delaunay triangulation., AIAA paper 91-0792.

[53] W.H. Frey and D.A. Field (1991), Mesh relaxation: a new technique for improving triangulations, Int. j. numer. methods eng. 31, 1121-1131.

[54] P.L. George et E. Séveno (1992), Génération de maillages par une méthode de type frontal, RR INRIA 1725.

[55] P. Möller and P. Hansbo (1995), On advancing-front mesh generation in three dimensions, Int. j. numer. methods eng. 38, 3551- 3569.

[56] H.K. Ruud and S.O. Wille (1990), An advancing front algorithm for three dimensional mesh generation, in G.N. Pandle and J. Middleton (eds), NUMETA 90, Numerical methods in engineering: Theory and applications, 2, 1141-1148, Elsevier.

[57] E. Seveno (1997), Towards an adaptive advancing front method, in 6th International Meshing Roundtable, Park City, Utah.

[58] R. Löhner ,(1996), Extensions and improvements of the advancing front grid generation technique, Commun. numer. methods eng. 12, 683-702.

[59] A. Le Pourhiet – Résolution numérique des équations aux dérivées partielles. Une première approche, Cépaduès, 1988.

[60] H. Jin and R.I. Tanner (1993), Generation of unstructured tetrahedral meshes by advancing front technique, Int. j. numer. methods eng. 36, 1805-1823.

[61] P.L. George and E. Séveno (1994), The advancing-front mesh generation method revisited, Int. j. numer. methods eng. 37, 3605-3619.

[62] SHARIFI, H, (1996), Maillage Automatique par la Méthode de quadtree/octree Modifiée . Thèse de doctorat, Université Laval.

[63] T.D. Blacker and M.B. Stephenson (1991), Paving, A new approach to automated quadrilateral mesh generation, Int. j. Numer, methods eng.32, 811-847.

[64] T.D. Blacker (1996), the cooper tool, in 5th International Meshing Roundtable, Pittsburgh, Pennsylvania, 13-29.

[65] M.A. Price, C.G. Armstrong and M.A. Sabin, (1995). Hexahedral mesh generation by medial surface subdivision, Part I, Solids with convex edges, Int. j. numer. methods eng. 38, 3335-3339.

[66] M.A. Price and C.G. Armstrong (1997), Hexahedral mesh generation by medial surface subdivision; Part II. Solids with flat and concave edges, Int. j. numer. Methods, eng.40,111-136.

[67] M.C. Rivara (1984), Mesh refinement processes based on the generalised bisection of simplices, SIAM J of Num. Ana. 21, 604-613.

[68] Haber R. H. , Shephard M. S. , Abel J. F., Gallagher R. H. and Greenberg D. P. (1985) "A general two dimensional finite element preprocessor utilizing discrete transfinite mapping", Int. j. numer. methods eng. , vol. 17 p1005-1044.

[69] Carter L., Wellford J. R. and Gorman M. R. (1985), A finite elemet transitional mesh generation procedure using sweeping function, Int. j. numer. methods eng. , vol. 26, p2623-2643, 1988.

[70] O. C. Zienkiewicz, (1977).The Finite Element Method, 3rd Edn. McGraw-Hill, New York .

[71] M. Bonnet et A. Frangi , 2007. Analyse des solides déformables par la méthode des éléments finis, Editions de l'Ecole polytechnique.

[72] R. D. Cook, D. S. Malkus et M. E. Plesha , 1989. Concepts and applications of finite element analysis, 3 éd., Wiley.

[73] G. Dhatt et G. Touzot , 1984. Une présentation de la méthode des éléments finis, Maloine.

[74] G. Dhatt, G. Touzot et E. Lefrancois , 2005. Méthode des éléments finis, Hermès.

[75] O. C. Zienkiewicz et R. L. Taylor , 1989. La méthode des éléments finis. Formulation de base et problèmes linéaires, AFNOR.

[76] J.-L. Batoz et G. Dhatt , 1990. Modélisation des structures par éléments finis, Volume 1. Solides élastiques, Hermès.

1.1 Introduction :

Depuis ses premières utilisations vers 1955, cette technique la Méthode des éléments finis n'a cessé de progresser et de susciter l'intérêt d'un nombre important de chercheurs qui ont contribué à la rendre la plus adéquate dans plusieurs domaines.

La théorie et l'application de cette méthode pour l'analyse linéaire ou non linéaire des structures sont traitées dans plusieurs ouvrages, entre autres DHATT & TOUZOT, GALLAGHER, YANG et ZIENKIEWICZ. Un bref exposé de la méthode est présenté dans ce qui suit. L'application pour l'étude du comportement linéaire et non linéaire de la torsion des barres fera l'objet du reste du chapitre.

Généralement, il existe plusieurs procédures d'analyse par éléments finis dépendantes de la forme que prend l'équation aux dérivées partielles. Pour chaque procédure utilisée, la méthode des éléments finis permet de passer d'une certaine forme de l'équation aux dérivées partielles à un système d'équations algébriques à résoudre tout en discrétisant le milieu étudié.

Pour le cas linéaire, la résolution du système se fait par des méthodes classiques de résolution des systèmes algébriques **(méthode de Gauss, de Choleskii ou Crout).**

Un problème d'élasticité est résolu si l'on connaît le vecteur déplacement en tout point du solide. Notre **problème** est donc défini par un **nombre infini** de paramètres. Un tel **système** est dit **continu**. La méthode des éléments finis remplace le système continu par un modèle discret caractérisé par un nombre fini de paramètres.

Les principales étapes de la méthode des éléments finis sont :

1. La géométrie est décomposée en domaines de forme géométrique simple (les éléments) relies entre eux en des points appelés nœuds. L'élément utilisé sera le **triangle à trois nœuds.**

2. Le champ de déplacements dans chaque élément est défini en fonction des **déplacements des nœuds de l'élément**. On en déduit l'état de **déformation** et l'état de **contrainte** en tout **point** de l'**élément** ainsi que **l'énergie de déformation** de l'élément et sa **matrice de rigidité**.

3. La matrice de rigidité globale est construite à partir des matrices de rigidité élémentaires.

4. Après mise en place des conditions aux limites et des charges, on calcule les déplacements inconnus puis, dans chaque élément, les déformations et les contraintes.

Les déplacements nodaux sont les inconnues du problème. Cette méthode qui consiste à prendre les déplacements nodaux comme inconnues du problème s'appelle la **méthode des déplacements.**

1.2 Maillage de la structure :

Le solide à étudier est décomposé en triangles à trois nœuds (figure 1.1). Cette opération s'appelle maillage. Il en résulte deux tables :

- la table des nœuds : elle contient les coordonnées des nœuds.
- la table des éléments : pour chaque élément, elle contient les numéros des trois nœuds (dans le sens trigonométrique), les caractéristiques du matériau, l'épaisseur…

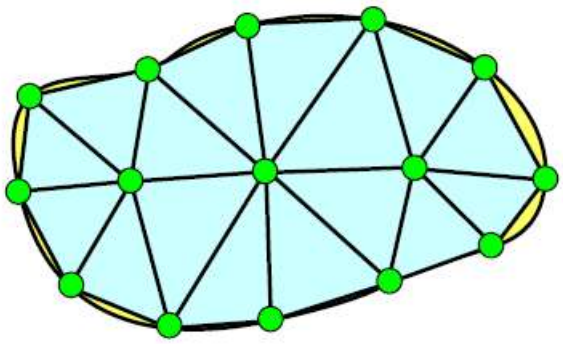


Figure 1.1 Décomposition d'un élément de structure en éléments finis

L'opération de décomposition est le **maillage**. Un bon maillage doit faciliter les opérations prévues par un programme d'ordinateur. Il faut donc adopter une méthode systématique de numérotation des éléments et des nœuds.

1.2.1Méthodes :

Deux méthodes sont en usage. Il en existe d'autres mais celles-ci sont les plus répandues. On en donnera un exemple pour un problème à deux dimensions.

1.2.1.1Première méthode

Les nœuds sont numérotés à la suite, de gauche à droite (ou de bas en haut) dans un système dextrogyre.

Une numérotation distincte est utilisée pour les éléments. Elle est de même nature que la précédente **(figure1.2).** On peut établir des tableaux donnant les numéros des nœuds appartenant à un élément (**par exemple** : élément 3 ; nœuds 3, 4, 8, 9). Ce système est le plus répandu [71].

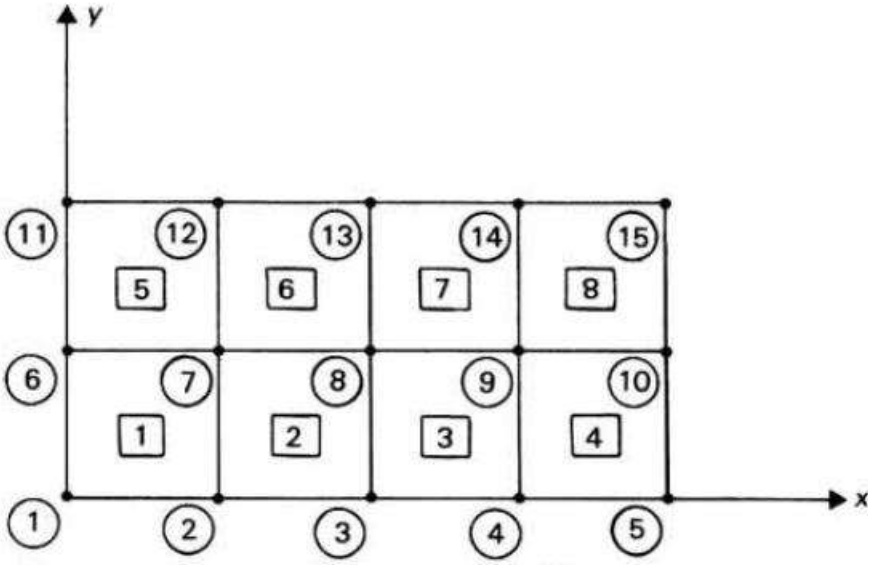


Figure 1.2 – Numérotation des éléments et des nœuds

1.2.1.2 Seconde méthode :

Chaque nœud et chaque élément sont identifiés par une paire ordonnée de nombres entiers ou indices **(figure 1.3).**

- **Nœuds** : le premier nombre désigne la ligne, le second celui de la colonne.

- **Éléments** : l'affectation des numéros se fait de la même manière. Chaque numéro a la même valeur que son nœud *sud-ouest*, soit le plus près de l'origine. **Exemple :** élément 12, nœud *sud-ouest* (1.2) [72].

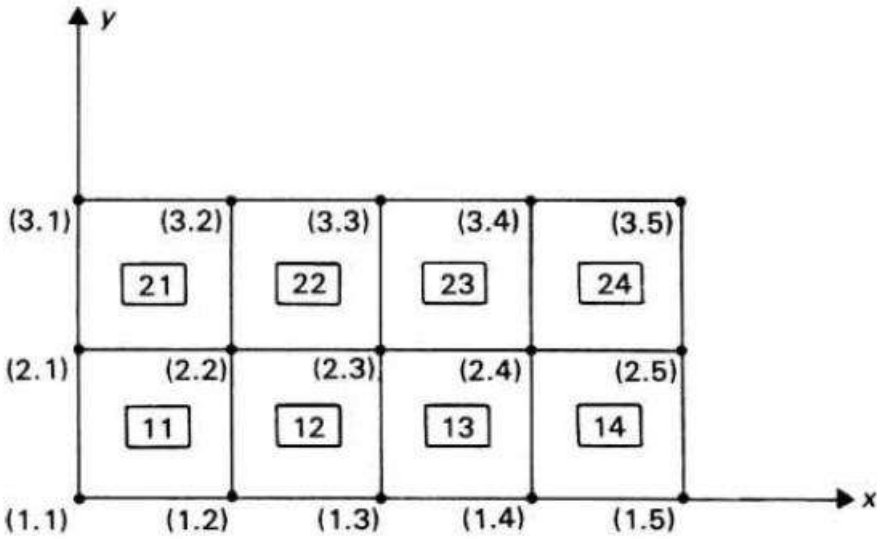


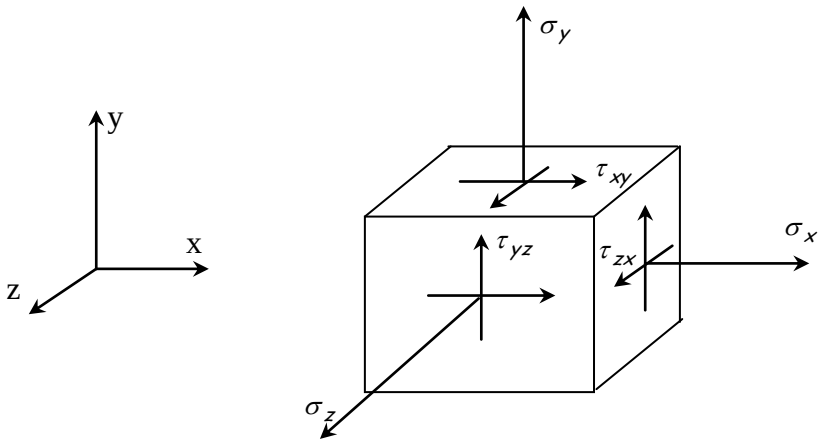
Figure 1.3 – Autre méthode de numérotation des éléments et des nœuds.

1.3 Revue de la théorie de Base :

En général, les contraintes et les déformations dans une structure consistent en six composantes

σ_x, σ_y, σ_z, τ_{xy}, τ_{yz}, τ_{zx} pour les contraintes

Et ε_x, ε_y, ε_z, γ_{xy}, γ_{yz}, γ_{zx} pour les déformations.



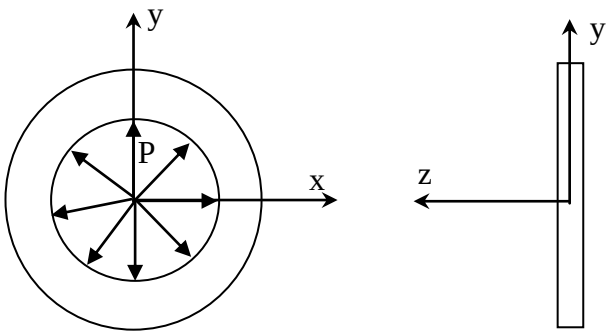
Dans certains cas, l'état de contraintes et de déformations pour être simplifie. Une analyse générale de la structure 3-D peut alors être réduite à une analyse 2-D.

1.3-1 Les Problèmes Plans 2-D :

a) Contrainte Plane :

$$\sigma_x = \tau_{yz} = \tau_{zx} = 0. \text{ et } \varepsilon_z \neq 0. \qquad (1.1)$$

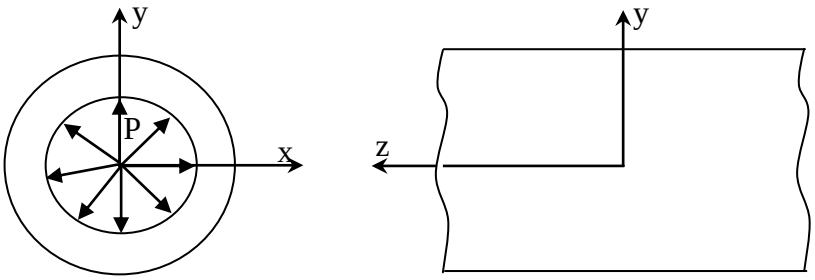
Une structure plane épaisse (thin) avec épaisseur constant et la charge est dans le plan dans la structure (x,y).



b) Déformation Plane :

$$\varepsilon_x = \gamma_{yz} = \gamma_{zx} = 0 \qquad (\sigma_z \neq 0.) \tag{1.2}$$

Une structure longue avec une section et le chargement est le long de la longueur (direction z).



1. 3.2 Les relations constitutives (contrainte – Déformation – Température) :

Pour les matériaux élastiques et isotopiques, on a :

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} 1/E & -\upsilon/E & 0 \\ -\upsilon/E & 1/E & 0 \\ 0 & 0 & 1/E \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} + \begin{Bmatrix} \varepsilon_{x0} \\ \varepsilon_{y0} \\ \gamma_{xy0} \end{Bmatrix} \tag{1.3}$$

Où $\vec{\varepsilon} = [E]^{-1}.\vec{\sigma} + \vec{\varepsilon}_0$

Où $\vec{\varepsilon}$ est la déformation initiale, E est le module d'young, ν le rapport de Poisson et le module de cisaillement G. Notons que,

$$G = \frac{E}{2(1+\nu)} \tag{1.4}$$

Ce qui signifie qu'il y a deux constantes indépendantes des matériaux homogènes et isotropes.

On peut aussi exprimer les contraintes en terme de déformation par la résolution de l'équation suivante :

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E}{1-\upsilon^2} \begin{bmatrix} 1 & \upsilon & 0 \\ \upsilon & 1 & 0 \\ 0 & 0 & \frac{1-\upsilon}{2} \end{bmatrix} \left(\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} - \begin{Bmatrix} \varepsilon_{x0} \\ \varepsilon_{y0} \\ \gamma_{xy0} \end{Bmatrix} \right) \tag{1.5}$$

Où $\vec{\sigma} = [E]\vec{\varepsilon} + \vec{\sigma}_0$

$\vec{\sigma}_0 = -E\vec{\varepsilon}_0$ est la contrainte initiale

Ces relations sont valides pour le cas de la contrainte plane. Pour le cas des la déformation plane, on a remplacé les contraintes des matériaux dans les équations ci-dessus dans la manière suivante :

$$E \to \frac{E}{1-\upsilon^2},\ \upsilon \to \frac{\upsilon}{1-\upsilon^2} \text{ et} \qquad G \to G \tag{1.6}$$

Par exemple, la contrainte est relie aux déformations par :

$$\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{Bmatrix}=\frac{E}{(1+\upsilon)(1-2\upsilon)}\begin{bmatrix}1-\upsilon & \upsilon & 0\\ \upsilon & 1-\upsilon & 0\\ 0 & 0 & \frac{1-2\upsilon}{2}\end{bmatrix}\left(\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{Bmatrix}-\begin{Bmatrix}\varepsilon_{x0}\\ \varepsilon_{y0}\\ \gamma_{xy0}\end{Bmatrix}\right)$$

Dans le cas de la déformation plane.

Les déformations initiales due aux changements de température (chargement thermique) est donnée par :

$$\begin{Bmatrix}\varepsilon_{x0}\\ \varepsilon_{y0}\\ \gamma_{xy0}\end{Bmatrix}=\begin{Bmatrix}\alpha\Delta T\\ \alpha\Delta T\\ 0\end{Bmatrix} \tag{1.7}$$

Où α est le coefficient de dilatation thermique, ΔT est la variation de la température.

Notons que si la structure est libre de déformer sur le changement thermique, il y'aura pas de contraintes élastiques dans la structure [73].

1.3.3 Les relations Déformations – Déplacements :

Pour des déformations petites et des rotations petites, on a :

$$\varepsilon_x=\frac{\partial u}{\partial x},\qquad \varepsilon_y=\frac{\partial v}{\partial y},\qquad \gamma_{xy}=\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x},$$

Dans une forme matricielle,

$$\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{Bmatrix}=\begin{bmatrix}\partial/\partial x & 0\\ 0 & \partial/\partial y\\ \partial/\partial y & \partial/\partial x\end{bmatrix}\begin{Bmatrix}u\\ v\end{Bmatrix},\qquad \text{où } \vec{\varepsilon}=[D]\vec{u} \tag{1.8}$$

De cette équation, on connaît que les déplacements (et alors les contraintes) sont d'ordre inférieur que les déplacements, si les déplacements sont représentés par les polynômes [74].

1.3.4 Equations d'équilibre :

Dans la **théorie d'élasticité**, les contraintes dans les structures doivent satisfaire les équations d'équilibre suivant :

$$\frac{\partial\sigma_x}{\partial x}+\frac{\partial\tau_{xy}}{\partial y}+f_x=0, \tag{1.9}$$

$$\frac{\partial\tau_{xy}}{\partial x}+\frac{\partial\sigma_y}{\partial y}+f_y=0,$$

où f_x et f_y sont des forces de poids (tels que le forces de gravitation) par unité de volumes.

Dans M.E.F, ces conditions d'équilibres sont satisfaites approximativement [75].

1.1.5 Les conditions d'équilibres :

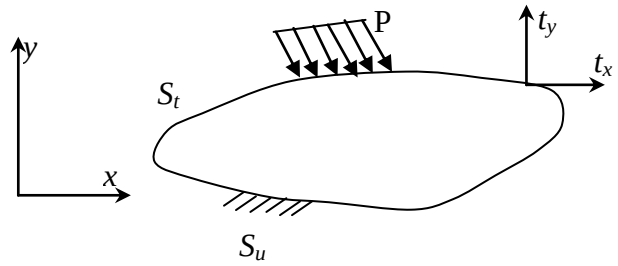
La frontière S du corps peut être di6sée en deux partie S_u et S_t.

Les conditions aux limites (C.L)

sont décrit par,

$u = \bar{u}$, $v = \bar{v}$ sur S_u

$t_x = \bar{t}_x$, $t_y = \bar{t}_y$ sur S_t



Dans lequel t_x et t_y sont les forces de traction (contraintes sur la frontière) et les quantités avec les barres sont les valeurs connues.

Dans M.E.F, tous les types de charges (les charges surfaciques distribuées, les forces volumiques, les forces concentrées et les moments, etc. …) sont convertis aux forces appliquées aux nœuds.

1.3.6 Solution d'élasticité Exacte :

La solution exacte (déplacements, déformations et contraintes) d'un problème donnée doit satisfaire l'équations d'équilibres (9), les conditions d'équilibres (10) et les conditions de compatibilité (la structure doit déformer d'une manière continue).

1.4Les Eléments Finis pour les problèmes 2-D :

1.4.1 Une formule Générale pour la matrice de rigidité :

Les déplacements (u, υ) dans un élément plan sont interpolés à partir des déplacements nodaux (u_i, υ_i) en utilisant les fonctions de formes N_i comme suit:

$$\begin{Bmatrix} u \\ \upsilon \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & \cdots \\ 0 & N_1 & 0 & N_2 & \cdots \end{bmatrix} \begin{Bmatrix} u_1 \\ \upsilon_1 \\ u_2 \\ \upsilon_2 \\ \vdots \end{Bmatrix} \quad ou \quad \vec{u} = [N]\vec{d} \tag{1.11}$$

Où $[N]$ est la matrice de fonctions de forme $\vec{u}$ est le vecteur déplacement et $\vec{d}$ est le vecteur déplacement nodal.

Ici, nous avons supposé les valeurs nodales que u dépend des valeurs nodales de u seulement et v dépend des valeurs nodales dυ seulement.

A partir de la relation déformation déplacement (Eq. (8)), le vecteur déformation est :

$$\vec{\varepsilon} = [D]\vec{u} = [D][N]\vec{d} \qquad ou \qquad \vec{\varepsilon} = [B]\vec{d} \tag{1.12}$$

Où $[B] = [D][\vec{N}]$ est la matrice de déformation déplacement.

Considérons **l'énergie de déformation stockée dans l'élément**,

$$U = \frac{1}{2}\int_v \sigma^T \vec{\varepsilon}.dv = \frac{1}{2}\int_v \left(\sigma_x \varepsilon_x + +\sigma_y \varepsilon_y + \tau_{xy}\gamma_{xy}\right)dv$$

$$U = \frac{1}{2}\int_v (E\vec{\varepsilon})^T \vec{\varepsilon} dv = \frac{1}{2}\int_v \varepsilon^T [E]\varepsilon\, dv = \frac{1}{2} d^T \int_v [B]^T [E][B] dv \quad \vec{d} = \frac{1}{2}\vec{d}^T [K]\vec{d}$$

A partir de ceci, on obtient la **formule générale** pour **la matrice de rigidité de l'élément,**

$$[K] = \int_v [B]^T [E][B] dv \tag{1.13}$$

Notons que contraint au cas 1-D, [E] ici est une matrice qui est donnée par la relation contrainte Déformation.

La matrice de rigidité [K] définir par (13) symétrique si [E] est symétrique.
Aussi notant que pour les propriétés du matériau donné, le comportement de [K] dépend de [B] seulement, qui dépend des fonctions de formes.

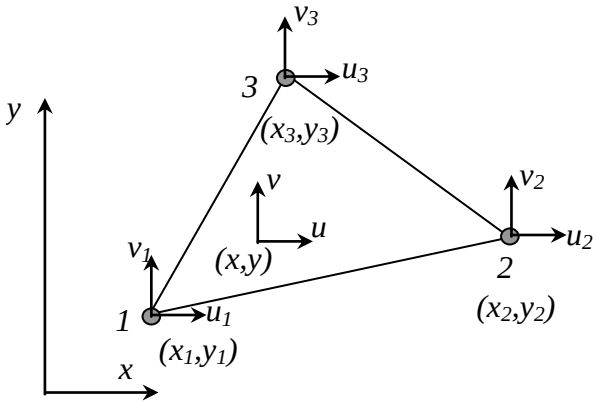
Alors, la qualité des éléments finis dans la représentation du comportement de la structure est enterrement déterminé par le choix des fonctions de forme.

La plus part des éléments 2-D généralement employés sont linéaires ou quadratique, triangulaire et quadrilatérales [76].

1.4.2 Elément Triangle à déformation constante (CST ou T3) :

Ceci est l'élément 2-D le plus simple, qui est aussi appelé élément triangulaire linéaire.

Pour cet élément, on a trois nœuds aux sommets du triangle, qui sont numérotés dans la direction antihoraire.



Chaque nœud à deux degrés de liberté (peut se déplacer dans les directions x et y.) Les déplacements u et υ sont supposés être linaires dans l'élément.

$$u = b_1 + b_2x + b_3y \qquad \text{et} \qquad v = b_4 + b_5x\, b_6y \tag{1.14}$$

Où b_i (i = 1, 2,…6) sont des constants. A partir de ceci, les déformations sont trouvées égale à :

$$\varepsilon_x = b_2,\ \varepsilon_y = b_6,\ \gamma_{xy} = b_3 + b_5 \tag{1.15}$$

Qui sont constante à travers l'élément. Aussi, on a appelé cet élément CST.

Les déplacements donnés par (14) doivent satisfaire les six équations suivantes :

$u_1 = b_1 + b_2x_1 + b_3y_1$

$u_2 = b_1 + b_2x_2 + b_3y_3$

……..

$v_3 = b_4 + b_5x_3 + b_6y_3$

Résolvant ces équations, on peut trouver les coefficients b_1, b_2,…, et b_6 en fonctions des déplacements et les coordonnées nodales.

Substituant ces coefficients dans (14) et réarrangeant les termes, on obtient :

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix} \tag{1.16}$$

Où les fonctions de forme (fonctions linéaires de x et y) sont :

$$\begin{aligned} N_1 &= \frac{1}{2A}\{(x_2y_3 - x_3y_2) + (y_2 - y_3)x + (x_3 - x_2)y_1\} \\ N_2 &= \frac{1}{2A}\{(x_3y_1 - x_1y_3) + (y_3 - y_1)x + (x_1 - x_3)y\} \\ N_3 &= \frac{1}{2A}\{(x_1y_2 - x_2y_1) + (y_1 - y_2)x + (x_2 - x_1)y\} \end{aligned} \tag{1.17}$$

et

$$A = \frac{1}{2}\det\begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \tag{1.18}$$

est la surface du triangle (Démontrer ceci!)

Utilisant la relation déformation-déplacement (8), les résultats (16) et (17), on obtient,

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = [B]\vec{d} = \frac{1}{2A}\begin{bmatrix} y_{23} & 0 & y_{31} & 0 & y_{12} & 0 \\ 0 & x_{32} & 0 & x_{13} & 0 & x_{21} \\ x_{32} & y_{23} & x_{13} & y_{31} & x_{21} & y_{12} \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix} \tag{1.19}$$

Où $x_{ij} = x_i - x_j$ et $y_{ij} = y_i - y_j$ (i,j = 1,2,3). Une autre fois, on voit que les déformations constantes dans l'élément. De relation contrainte-déformation (Eq.(5)), o, voit que les contraintes obtenues utilisant l'élément CST sont aussi constantes.

Appliquant la formule (13), on obtient la matrice de rigidité de l'élément pour l'élément CST.

$$[K] = \int_v [B]^T [E][B] dv = tA\left([B]^T [E][B]\right) \qquad (1.20)$$

Dans laquelle t est l'épaisseur de l'élément. Notant que [K] pour le CST est une matrice symétrique 6x1. La multiplication des matrices dans (20) peut être exécuté par un programme informatique.

Les deux expressions des fonctions de forme dans (17) et leurs dérivés sont longues.

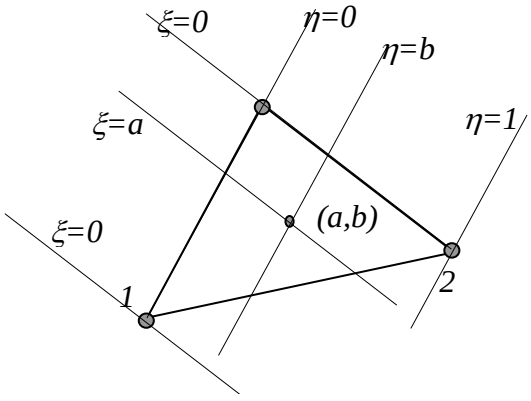
On introduit les coordonnées naturelles (ξ,η) dans le triangle, alors les fonctions de forme peut être représenter simplement par [75]:

$$N_1 = \xi, \qquad N_2 = \eta, \qquad N_3 = 1 - \xi - \eta \qquad (1.21)$$

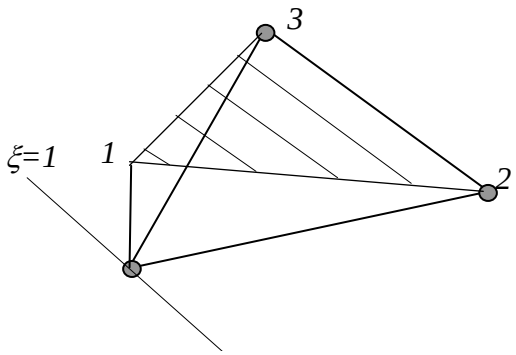
Notons que, $N_1 + N_1 + N_1 = 1$ (1.22)

Comme dans le cas 1-D, $N_1 = \begin{cases} 1 & \text{au noeud i} \\ 0 & \text{à d'autre noeud} \end{cases}$ (1.23)

Le tracé de la fonction de forme N_1 est montré dans la figure suivante. N_2 et N_3 ont des caractéristiques similaires.



Les coordonnées naturelles



La fonction de forme N_1

Nous avons deux systèmes de coordonnées pour l'élément ; les coordonnées globales (x,y) et coordonnées naturelles (ξ,η). La relation entre deux est données par

$$\begin{aligned} x &= N_1 x_1 + N_2 x_2 + N_3 x_3 \\ y &= N_1 y_1 + N_2 y_2 + N_3 y_3 \end{aligned} \qquad (1.24)$$

où,

$$\begin{aligned} x &= x_{13}\xi + x_{23}\eta + x_3 \\ y &= y_{13}\xi + y_{23}\eta + y_3 \end{aligned}$$

Où $x_{ij} = x_i - x_j$ et $y_{ij} = y_i - y_j$ (i,j = 1,2,3) comme définie précédemment.

Le déplacement u ou η dans l'élément peut être fonctions de (x,y) ou (ξ,η).

$$\text{On a } \begin{Bmatrix} \frac{\partial u}{\partial \xi} \\ \frac{\partial u}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{Bmatrix} = [J] \begin{Bmatrix} \frac{\partial u}{\partial \xi} \\ \frac{\partial u}{\partial \eta} \end{Bmatrix} \tag{1.26}$$

Où [J] est appelée la matrice du Jacobien de transformation

$$J = \begin{bmatrix} x_{13} & y_{13} \\ x_{23} & y_{23} \end{bmatrix} \qquad \left[J^{-1}\right] = \frac{1}{2A} \begin{bmatrix} y_{23} & -y_{13} \\ -x_{23} & x_{13} \end{bmatrix} \tag{1.27}$$

où det[J] = $x_{13}\, y_{23} - x_{23}\, y_{13} = 2A$

De (26), (27), (16), (21) on a :

$$\begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{Bmatrix} = \frac{1}{2A} \begin{bmatrix} y_{23} & -y_{13} \\ -x_{23} & x_{13} \end{bmatrix} \begin{Bmatrix} \frac{\partial u}{\partial \xi} \\ \frac{\partial u}{\partial \eta} \end{Bmatrix} = \frac{1}{2A} \begin{bmatrix} y_{23} & -y_{13} \\ -x_{23} & x_{13} \end{bmatrix} \begin{Bmatrix} u_1 - u_3 \\ u_2 - u_3 \end{Bmatrix} \tag{1.28}$$

D'une façon similaire,

$$\begin{Bmatrix} \frac{\partial \upsilon}{\partial x} \\ \frac{\partial \upsilon}{\partial y} \end{Bmatrix} = \frac{1}{2A} \begin{bmatrix} y_{23} & -y_{13} \\ -x_{23} & x_{13} \end{bmatrix} \begin{Bmatrix} \upsilon_1 - \upsilon_3 \\ \upsilon_2 - \upsilon_3 \end{Bmatrix} \tag{1.29}$$

Utilisant les résultats dans (28) et (29), et les relations

$\vec{\varepsilon} = D\, \vec{u} = D\ \ N\, \vec{d} = B\, \vec{d}$, on obtient la matrice déplacement-déformation,

$$[B] = \frac{1}{2A} \begin{bmatrix} y_{23} & 0 & y_{31} & 0 & y_{12} & 0 \\ 0 & x_{32} & 0 & x_{13} & 0 & x_{21} \\ x_{32} & y_{23} & x_{13} & y_{31} & x_{21} & y_{12} \end{bmatrix} \tag{1.30}$$

Ce qui est la même expression que celle dérivée dans (1.19).

Application de l'élément CST [71]:

- Il est utilisé dans les surfaces où le gradient de déformation est petit,
- Utilisé dans les surfaces de transition de maillage (maillage fin au maillage gros),
- E6ter d'utiliser CST dans la concentration de contraintes et d'autre surface critique de la structure,

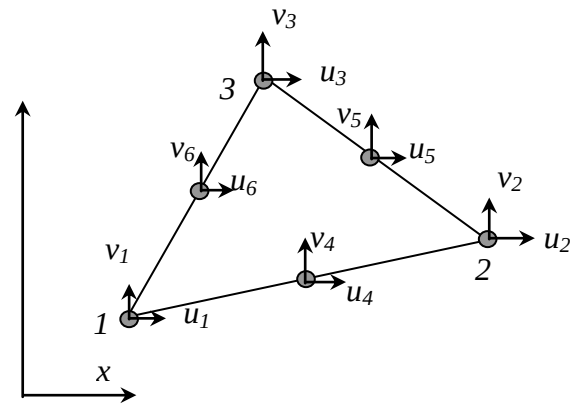
- Recommandé pour une analyse préliminaire d'E.F. des problèmes 2-D.

1.4.3 Triangle de déformation linéaire (LST ou T6) [73] :

Cet élément est aussi appelé élément triangulaire quadratique.

Il y'a six nœuds dans cet élément.

Chaque nœud à 2 degré de liberté. Les déplacements (u,v) sont supposés une fonction quadratique de (x,y).



$$u = b_1 + b_2x + b_3y + b_4x^2 + b_5xy + b_6y^2$$
$$v = b_7 + b_8x + b_9y + b_{10}x^2 + b_{11}xy + b_{12}y^2 \qquad (1.31)$$

Ou b_i (i=1,2,…,12) sont des constantes. De ceci, les déformations sont trouvées :

$$\varepsilon_x = b_2 + 2b_4x + b_5x$$
$$\varepsilon_x = b_9 + b_{11}x + 2b_{12}x \qquad (1.32)$$
$$\gamma_{xy} = (b_3 + b_8) + (b_5 + 2b_{10})x + (2b_6 + b_{11})y$$

Ce sont des fonctions linéaires. Ainsi, on a le triangle de déformations linéaires (LST), qui fournit des meilleurs résultats que le CST.

Dans le système de coordonnées naturelles, les fonctions de forme pour l'élément LST sont :

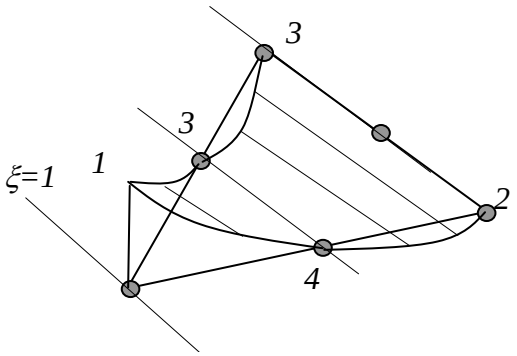
$$N_1 = \xi(2\xi - 1) \qquad N_2 = \eta(2\eta - 1) \qquad N_3 = \xi(2\xi - 1)$$
$$N_4 = 4\xi\eta \qquad N_5 = 4\eta\xi \qquad N_6 = 4\xi\xi \qquad (1.33)$$

Dans lequel $\zeta = 1 - \xi - \eta$ chaque de ces six fonctions de forme représente une forme quadratique dans l'élément comme montré dans la figure.

Les déplacements peut être écrit comme:

$$u = \sum_{i=1}^{6} N_i u_i \qquad v = \sum_{i=1}^{6} N_i v_i \qquad (1.34)$$

La matrice de rigidité de l'élément est toujours donnée par : $K = \int_v B^T E B \, dv$ mais ici $[B]^T [E][B]$ est quadratique en x et y.

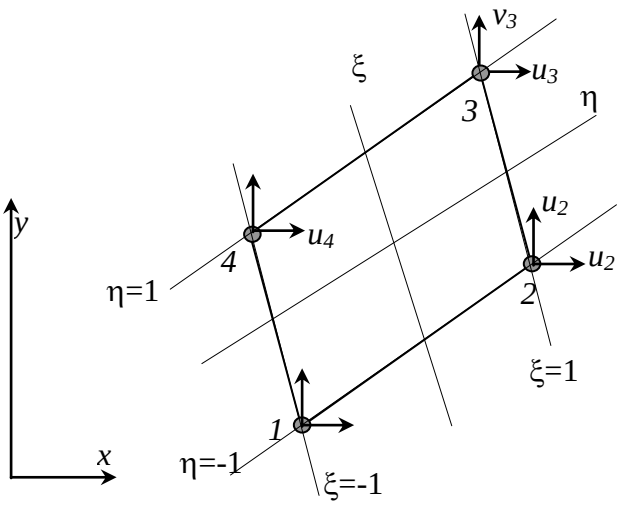


La fonction de forme N_1 pour LST.

1.4.4 Elément quadrilatéral linéaire (Q_4) [74] :

Il y'a quatre nœuds au sommets de la forme quadrilatéral.

Dans le système de coordonnées naturelles (ξ,η), les quatre fonctions de la forme sont :



$$N_1=\frac{1}{4}(1-\xi)(1-\eta) \quad N_2=\frac{1}{4}(1+\xi)(1-\eta)$$
$$N_3=\frac{1}{4}(1+\xi)(1+\eta) \quad N_4=\frac{1}{4}(1-\xi)(1+\eta) \qquad (1.35)$$

Notant qu'à n'importe quel point à l'intérieure de la de l'élément, comme espérer.

Le champ de déplacement est donnée par :

$$u=\sum_{i=1}^{4} N_i u_i \qquad v=\sum_{i=1}^{4} N_i v_i \qquad (1.36)$$

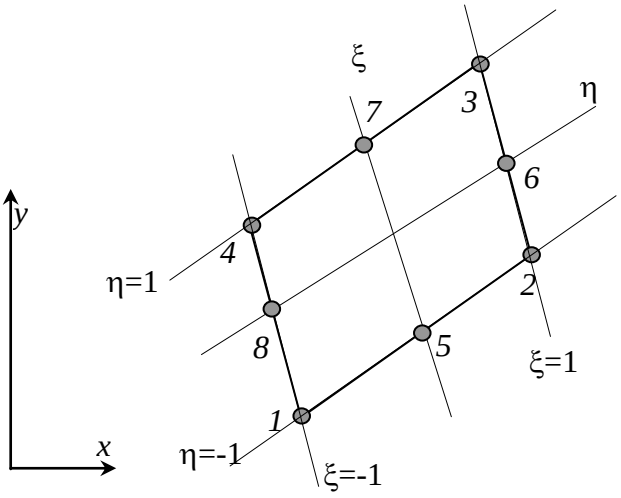
Ce sont des fonctions bilinéaires à travers l'élément.

1.4.5 Elément quadrilatéral quadratique (Q8) [71]:

C'est le plus largement utilisé des éléments pour les problèmes 2-D du à sa haute précision dans l'analyse.

Il y'a huit nœuds pour cet élément, quatre sommets et quatre au milieu.

Dans le système de coordonnées naturelles (ξ,η), les huit fonctions de forme sont,



$$N_1 = \frac{1}{4}(1-\xi)(1-\eta)(\xi+\eta+1) \qquad N_3 = \frac{1}{4}(1+\xi)(1+\eta)(\xi+\eta-1)$$
$$N_2 = \frac{1}{4}(1+\xi)(1-\eta)(\eta-\xi+1) \qquad N_4 = \frac{1}{4}(1-\xi)(1+\eta)(\xi-\eta+1)$$
$$N_5 = \frac{1}{2}(1-\eta)(1-\xi^2) \qquad N_7 = \frac{1}{2}(1+\eta)(1-\xi^2) \qquad (1.37)$$
$$N_6 = \frac{1}{2}(1+\xi)(1-\eta^2) \qquad N_8 = \frac{1}{2}(1-\xi)(1-\eta^2)$$

Une autre foie, on a $\sum_{i=1}^{8} N_i = 1$ à n'importe quel point de l'élément.

Le champ de déplacement est donnée par :

$$u = \sum_{i=1}^{8} N_i u_i \qquad v = \sum_{i=1}^{8} N_i v_i \qquad (1.38)$$

Ce sont des fonctions quadratiques à travers l'élément. Les déformations et les contraintes à travers l'élément quadratique sont des fonctions linéaires, qui sont des bonnes représentations.

1. 4.6 Remarques :

Q_4 et T_3 sont généralement utilisés ensembles dans un maillage avec des éléments linéaires.

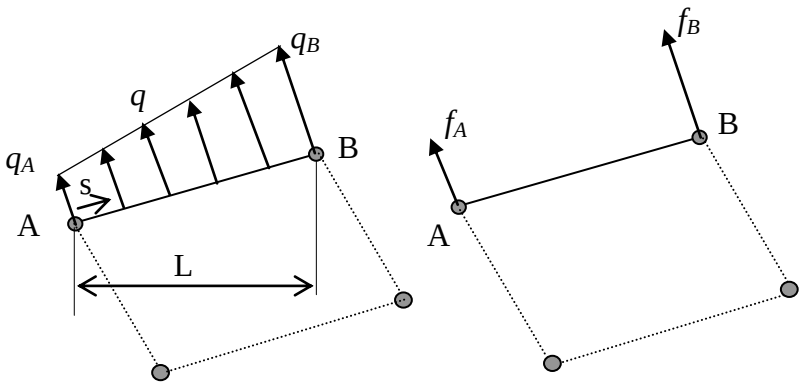
Q_8 et T_6 sont généralement appliqués dans un maillage composé avec des éléments quadratiques.

Les éléments quadratiques sont préférés pour l'analyse des contraintes, à cause de leurs hautes précision et leur flexibilité dans la modélisation des géométries complexes, tels que les frontières courbées.

1.4.7 Transformation de Charges [74] :

Les charges concentrées (forces), traction (charges de pression), et les forces volumique (poids) sont les principaux types de charges appliquées à la structure. Les deux types de charges, traction et les forces volumiques, doivent converties aux forces nodales en E.F.. Ainsi, elles ne peuvent pas être appliquées directement.

La conversion de ces charges est basée sur la même idée (concept du travail équivalent) que nous avons



utilisée pour les cas des éléments barres et poutres.

Supposant, par exemple on a une traction linéaire q en un élément Q4, comme montré dans la figure.

La traction est normale à la frontière. Utilisant la coordonnée locale (tangentiel) s, on peut écrire **le travail** fait par la **force de traction q** comme : $W_q = t\int_0^L u_n(s)q(s)ds$

Où t est l'épaisseur, L la longueur et u_n la composante du déplacement normale au côté AB.

Pour l'élément Q_n (champ de déplacement linéaire), on a $u_M(s) = \left(1-\frac{s}{L}\right)u_{MA} + \frac{s}{L}u_{MB}$. La force de traction q(s), qui est aussi linéaire, est donnée d'une manière similaire, $q(s) = \left(1-\frac{s}{L}\right)q_A + \frac{s}{L}q_B$.

Alors, on a : $W_q = t\int_0^L \left(\begin{bmatrix} u_{nA} & u_{nB} \end{bmatrix} \begin{bmatrix} 1-\frac{s}{L} \\ \frac{s}{L} \end{bmatrix} \right) \left(\begin{bmatrix} 1-\frac{s}{L} & \frac{s}{L} \end{bmatrix} \begin{bmatrix} q_A \\ q_B \end{bmatrix} \right) ds$

$$= \begin{bmatrix} u_{nA} & u_{nB} \end{bmatrix} t\int_0^L \begin{bmatrix} \left(1-\frac{s}{L}\right)^2 & \left(\frac{s}{L}\right)\left(1-\frac{s}{L}\right) \\ \left(\frac{s}{L}\right)\left(1-\frac{s}{L}\right) & \left(\frac{s}{L}\right)^2 \end{bmatrix} ds$$

et le vecteur force nodale équivalent est, $\begin{Bmatrix} f_A \\ f_B \end{Bmatrix} = \frac{tL}{3}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\begin{Bmatrix} q_A \\ q_B \end{Bmatrix}$

Notant pour q constant, on a : $\begin{Bmatrix} f_A \\ f_B \end{Bmatrix} = \frac{qtL}{3}\begin{Bmatrix} 1 \\ 1 \end{Bmatrix}$

Pour les éléments quadratiques (triangulaire ou quadrilatéral), la force de traction est converti aux forces au trois nœuds au lieu de deux nœuds.

1.4.8 Calcul des contraintes [72] :

La contrainte dans un élément est déterminée par la formule suivante :

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = [E]\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = [E][B]\vec{d}$$

Où [B] est la matrice déformation-déplacement et $\vec{d}$ est le vecteur déplacement nodal qui est connu pour chaque élément lorsque l'équation globale E.F. est résolue.

Les contraintes peuvent être évaluées à n'importe quels points dans l'élément (par exemple au centre) où aux nœuds. Les graphes des contours sont généralement utilisés dans les Logiciels A.E.F (durant Post-Processing) les utilisateurs pour 6sualiser les résultats des

contraintes.

La contrainte Von Mises [73] :

La contrainte Von Mises est la contrainte équivalente ou effective pour les contraintes 2-D ou 3-D. Pour un matériau ductile, le niveau de contrainte est acceptable si $\sigma_e \leq \sigma_y$

Où σ_e est la contrainte Von Mises et σ_y est la contrainte (yield) du matériau. C'est la généralisation des résultats expérimentaux 1-D et 2-D.

La contrainte Von Mises est définie par :

$$\sigma_e = \frac{1}{\sqrt{2}}\sqrt{(\sigma_1-\sigma_2)^2+(\sigma_2-\sigma_3)^2+(\sigma_3-\sigma_1)^2} \tag{1.40}$$

Dont lequel, et sont les trois contraintes principales au point considéré de la structure.

Pour des problèmes 2-D, les deux contraintes principales dans le plan sont déterminées par :

$$\begin{aligned}\sigma_1^p &= \frac{\sigma_x+\sigma_y}{2}+\sqrt{\left(\frac{\sigma_x-\sigma_y}{2}\right)^2+\tau_{xy}^2}\\ \sigma_2^p &= \frac{\sigma_x+\sigma_y}{2}-\sqrt{\left(\frac{\sigma_x-\sigma_y}{2}\right)^2+\tau_{xy}^2}\end{aligned} \tag{1.41}$$

Alors, on peut aussi exprimer la contrainte Von Mises en fonction des composantes de la contrainte x,y. Pour les conditions de contrainte plan, on a

$$\sigma_e = \frac{\sigma_x+\sigma_y}{2}\sqrt{(\sigma_x+\sigma_y)^2-3(\sigma_x\sigma_y-\tau_{xy}^2)} \tag{1.42}$$

Les contraintes moyennes :

Les contraintes sont généralement moyennées aux nœuds dans les logiciels pour fournir des valeurs précises.

Cette option doit être enlever aux nœuds entre deux matériaux ou dans des zones de géométries discontinues où il y'a des discontinuité de la contrainte.

1.4.9 Le savoir faire :

1) connaître les comportements de chaque type des éléments.

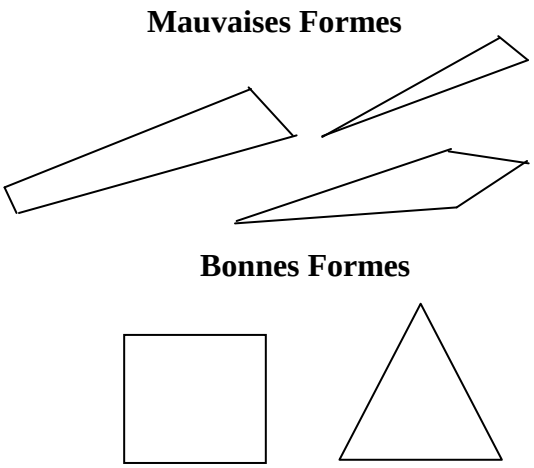
T_3 et Q4 : déplacement linéaire, déformation et contrainte constantes.

T_6 et Q_8 : déplacement quadratique, déformation et contrainte linéaires.

2) choisir le bon type des éléments pour un problème donné :

En doute, utiliser les éléments d'ordre supérieur ou un maillage fin.

3) choisir les éléments avec les rapports de forme grand et les angles des coins.



$$raport\ de\ \text{forme} = \frac{L_{max}}{L_{min}}$$

où L_{min} et L_{max} sont les longueurs caractéristiques minimales et maximales de l'élément respectivement.

4) Relier les éléments proprement

Ne laisse pas les espaces libres ou les éléments libres dans les modèles E.F.

1.5 La Méthode des éléments finis et l'industrie [72] :

Le calcul occupe une place stratégique avec les technologies de simulation (essais) dans le développement d'un produit complexe qui touche à différents domaines de la physique. Cela concerne les industries automobiles, navales, aéronautiques, ferroviaires, mais aussi les industries lourdes : centrales électriques, plates-formes pétrolières, et le génie civil.

Le calcul est indispensable lorsque l'on cherche à obtenir une solution optimisée pour réduire les coûts et les délais de fabrication. Grâce au calcul l'ingénieur peut tester plusieurs configurations pour optimiser le comportement d'un modèle à une prestation donnée. Cela évite de multiplier les prototypes et les essais tests réels, les supports physiques ne servent plus à chercher une solution, ils permettent de la valider. Attention, même précis un modèle ne fournis jamais qu'une approximation de la réalité, il est donc impossible de se passer des prototypes.

Le calcul s'applique aussi dans les domaines du « **process** ». Les procédés de fabrication tels que l'emboutissage, l'usinage grande vitesse, les dépôts de peinture, l'assemblage de tôlerie, la mise en forme des plastiques, peuvent être modélisés par éléments finis. Ici c'est une bonne représentation du comportement du phénomène physique qui sera recherchée pour pouvoir vérifier et valider un procédé de fabrication d'une pièce.

La méthode des éléments finis est de toutes les méthodes de discrétisation la plus utilisée car :

-elle peut traiter des problèmes de géométrie complexe ;

-elle couvre de nombreux domaines de la physique ;

-les moyens informatiques actuels (puissance des calculateurs, outils de visualisation) la rende facile de mise en œuvre.

L'introduction des données se fait par un fichier test4_1.m , il comporte:

```
%
%  Material Properties
%
  young =   210000.00 ;
  poiss =   0.28      ;
  denss =   0.0000    ;
  pstrs =   1         ;
  thick =  10.0       ;
%
% Coordinates
%
global coordinates
coordinates = [

00.00000       30.00000
22.50000       30.00000
45.00000       30.00000
67.50000       30.00000
90.00000       30.00000
00.00000       15.00000
22.50000       15.00000
45.00000       15.00000
67.50000       15.00000
90.00000       15.00000
00.00000       00.00000
22.50000       00.00000
45.00000       00.00000
67.50000       00.00000
90.00000       00.00000] ;

%
% Elements
%
global elements
elements = [
          6    7    2    1
          7    8    3    2
          8    9    4    3
          9   10    5    4
         11   12    7    6
         12   13    8    7
         13   14    9    8
         14   15   10    9
  ] ;
%
% Fixed Nodes
%
fixnodes = [
      1   , 1 ,    0.00000  ;
      1   , 2 ,    0.00000  ;
      6   , 1 ,    0.00000  ;
      6   , 2 ,    0.00000  ;
      11  , 1 ,    0.00000  ;
      11  , 2 ,    0.00000  ] ;
```

```
%
% Point loads
%
pointload = [ 10 , 2 , -10000.00] ;
%
% Side loads
%
sideload = [    ];
```

Le fichier des résultats test4_1.flavia.res comporte:

```
Gid Post Results File 1.0
###
# MAT_FEM  V.1.0
#
Result "Displacements" "Load Analysis"  1  Vector OnNodes
ComponentNames "X-Displ", "Y-Displ", "Z-Displ"
Values
     1         00000         00000
     2  4.43458e-002 -4.47608e-002
     3  7.69196e-002 -1.43535e-001
     4  9.63891e-002 -2.82475e-001
     5  1.02459e-001 -4.38873e-001
     6         00000         00000
     7  1.67948e-017 -4.00437e-002
     8  2.67055e-017 -1.41454e-001
     9  3.88450e-017 -2.80762e-001
    10  2.20285e-017 -4.40923e-001
    11         00000         00000
    12 -4.43458e-002 -4.47608e-002
    13 -7.69196e-002 -1.43535e-001
    14 -9.63891e-002 -2.82475e-001
    15 -1.02459e-001 -4.38873e-001
End Values
#
Result "Reaction" "Load Analysis"  1  Vector OnNodes
ComponentNames "Rx", "Ry", "Rz"
Values
     1 -3.00000e+004 7.10124e+003
     2        00000        00000
     3        00000        00000
     4        00000        00000
     5        00000        00000
     6 -2.36469e-011 -4.20248e+003
     7        00000        00000
     8        00000        00000
     9        00000        00000
    10        00000        00000
    11 3.00000e+004 7.10124e+003
    12        00000        00000
    13        00000        00000
    14        00000        00000
    15        00000        00000
```

```
End Values
#
Result "Stresses" "Load Analysis"  1  Matrix OnNodes
ComponentNames "Sx", "Sy", "Sz", "Sxy", "Syz", "Sxz"
Values
     1 4.49104e+002 1.25749e+002   0.0   -1.63190e+002  0.0   0.0
     2 3.69431e+002 3.74020e+001   0.0   -1.91355e+001  0.0   0.0
     3 2.54677e+002 4.21747e+001   0.0   -1.26784e+001  0.0   0.0
     4 1.22041e+002 1.02013e+001   0.0   -1.12485e+001  0.0   0.0
     5 7.01935e+001 4.83550e+001   0.0   -9.87726e+000  0.0   0.0
     6 1.56319e-013 3.55271e-014   0.0   -1.45993e+002  0.0   0.0
     7 1.17240e-013 2.20268e-013   0.0   -1.53419e+001  0.0   0.0
     8 6.66134e-014 2.22045e-014   0.0   -1.81561e+001  0.0   0.0
     9 9.76996e-014 -6.74127e-013   0.0   -1.87792e+001  0.0   0.0
    10 -3.31291e-013 1.21325e-012   0.0   -2.35937e+001  0.0   0.0
    11 -4.49104e+002 -1.25749e+002   0.0   -1.63190e+002  0.0   0.0
    12 -3.69431e+002 -3.74020e+001   0.0   -1.91355e+001  0.0   0.0
    13 -2.54677e+002 -4.21747e+001   0.0   -1.26784e+001  0.0   0.0
    14 -1.22041e+002 -1.02013e+001   0.0   -1.12485e+001  0.0   0.0
    15 -7.01935e+001 -4.83550e+001   0.0   -9.87726e+000  0.0   0.0
End Values
```

```
C    APPENDIX A. VARIABLE ORDER MESH GENERATION PROGRAM
C*****************************************************************
C*****************************************************************
C
C          VARIABLE ORDER MESH GENERATION PROGRAM
C
C*****************************************************************
C*****************************************************************
C
C          TITLE - INPUT TITLE
C          INRG OR NRG - NUMBER OF INPUT SUPERELEMENTS
C          INBP - NUMBER OF INPUT NODES
C          IPOL - NUMBER OF AUXILIARY POLAR SYSTEMS
C          ITMP - INPUT TEMPERATURE OPTION
C          ITHK - INPUT THICKNESS OPTION
C          IRENUM - RENUMBER OPTION
C          (A) 0 - NO RENUMBER
C          (B) 1 - RENUMBER
C          IPOLAR - OPTION OF COORDINATE SYSTEMS
C          (A) 0 - CARTESIAN COORDINATE SYSTEMS
C          (B) NUMBERS GREATER THAN 0 - DIFFERENT AUXILIARY POLAR SYSTEMS
C          NOP - ARRAY REPRESENTING NODE NUMBERS OF EVERY INPUT SUPERELEMENT
C          ITYPE - ARRAY REPRESENTING TYPE OF OUTPUT ELEMENT
C          (A) 1       3 NODE TRIANGULAR
C          (B) 2       4 NODE LINEAR
C          (C) 3       6 NODE ISOPARAMETRIC
C          (D) 4       6 NODE TRIANGULAR
C          (E) 5       8 NODE ISOPARAMETRIC
C          XP(100,1) - ARRAY REPRESENTING INPUT X - COORDINATE
C          YP(100,1) - ARRAY REPRESENTING INPUT Y - COORDIHATE
C          XP(100,2) - ARRAY REPRESENTING INPUT TEMPERATURE
C          YP(100,2) - ARRAY REPRESENTING INPUT THICKNESS, OR OTHER PROPERTIES
C          NROW1 - ARRAY REPRESENTING DIVISION IN ETA DIRECTION FOR EVERY
C          SUPERELEMENT
C          NCOL1 - ARRAY REPRESENTING DIVISION IN XI DIRECTION FOR EVERY
C          SUPERELEMENT
C          IND11 - BOUNDARY NODE INDICATOR OF EVERY INPUT SUPERELEMENT
C          NMAT - ARRAY REPRESENTING MATERIAL CODE OF EVERY INPUT SUPERELEMENT
C          NB - NUMBER OF GENERATED NODES
C          NUMLE OR NEL - NUMBER OF GENERATED ELEMENTS
C          NCN - ARRAY REPRESENTING NUMBER OF NODES ASSOCIATED WITH ANY
C          GENERATED ELEMENT
C          NNRB - ARRAY STORING GENERATED BOUNDARY NODES OF ALL SUPERELEMENT
C          IVOE - OPTION OF VARIABLE ORDER ELEMENT MESH
C          (A) 0       UNIFORM ORDER MESH
C          (B) 1       VARIABLE ORDER MESH
C          NJT - MAX. NUMBER OF GENERATED NODES
C          MEM - MAX. NUMBER OF GENERATED ELEMENTS
C          NM - MAX. NUMBER OF GENERATED ROWS OR COLUMNS IN A SUPERELEMENT
C          MAXCON - MAX. NUMBER OF ELEMENT A NODE CAN BE SHARED BY
C          IDIFF - MAX. NODE DIFFERENCE IN ALL GENERATED ELEMENTS
C          TO - OUTPUT EQUIPMENT NUMBER
C
c*********************************************************************
          COMMON XC(21,21),YC(21,21),TEMP(21,21),THK(21,21),NN(21,21),
          1NROWS,NCOL,IO,NCON(200,8),NUMEL,NB,NCN(200),IDIFF,IRENUM,JRN,
          2NM,NSE,MAXCON,MEM,NJT,MAT(200),NROW1(20),NCOL1(20),IND11(20),
          3NMAT(20),NOP(20,8),XP(100,2),YP(100,2),XNOD(441,2),YNOD(441,2),
          4INRG,ITYPE(20),JNT(441),MEMJT(10584),JMEM(441),NEWJT(441),
          5JOINT(441),INBP,ITMP,ITHK

          COMMON /TM/JTE(200),ITY(200),IVOE
          DIMENSION TITLE (20)
          DIMENSION ICROSS(20),NDUMY(20,5),XOFSET(20),YOFSET(20)
          DIMENSION IL(12)

      OPEN(UNIT=5,FILE="MAIL13.TXT",STATUS="OLD")
```

```
      OPEN(UNIT=6,FILE="OUTPUT.TXT",STATUS="UNKNOWN")

      IN=5
      IO=6
      DO 7 I=1,12
7     IL(I)=0

C-----------------------------------------------------------------
C     INPUT AND OUTPUT OF TITLT,CONTROL CARD,GLOBAL COORDINATES AND
C     CONNECTIVITY DATA
C-----------------------------------------------------------------
      NOFSET=20
      NSBN=100
      NM=21
      NSE=20
      NJT=441

      MEM=200
      DO 10 I=1,NOFSET
      XOFSET(I)=0.0
 10   YOFSET(I)=0.0
   WRITE(IO,20)
 20   FORMAT(1H1,17X,'------------'/18X,
  1'* MESH GENERATION PRAGROM *'/18X,
  2'---------------------------'///)
   WRITE(IO,30)NSE,NSBN,NOFSET,NJT,MEM
 30   FORMAT(10X,'CURRENT PROGRAM LIMITATIONS'/10X,
  1'MAX.NUMBER OF SUPUERELEMENTS=',I3/10X,
      2'MAX.NUMBER OF SUPERELEMENT NODES=',I4/10X,
      3'MAX.NUMBER OF AUXILIARY POLAR COORDINATE SYSTEMS=',I4/10X,
      4'MAX.NUMBER OF GENERATED NODES=',I4/10X,
      5'MAX.NUMBER OF GENERATED ELEMENTS=',I4///)
      READ(IN,40)TITLE
 40   FORMAT(20A4)
      READ(IN,*)   INRG,INBP,IPOL,IVOE,ITMP,ITHK,IRENUM
 50   FORMAT(9I5)
      IF(INRG.GT.NSE) WRITE (IO,60)
 60   FORMAT(//5X,'NUMBER OF INPUT REGIONS EXCEEDES ALLOWABLE',
  1   /5X,'PROGRAM TERMINATED')
      IF(INRG.GT.NSE) STOP
      IF(INBP.GT.NSBN) WRITE (IO,70)
 70   FORMAT(//5X,'NUMBER OF SUPERELEMENT NODES EXCEEDS ALLOWABLE',
      1     /5X,'PROGRAM TERMINATED')
      IF(INBP.GT.NSBN) STOP
      IF(IPOL.GT.NOFSET) WRITE (IO,80)
 80   FORMAT(//5X,'NUMBER OF AUXILIARY COORDINATE SYSTEMS EXCEEDS',
      1     /5X,'ALLOWABLE PROGRAM TERMINATED')
      IF(IPOL.GT.NOFSET) STOP
      MAXCON=24
      WRITE(IO,90)INRG,INBP,IPOL,IVOE,ITMP,ITHK,IRENUM
 90   FORMAT(/10X,'NUMBER OF INPUT REGIONS=',I3,
      1     /10X,'NUMBER OF INPUT NODES=',I4,
      2     /10X,'NUMBER OF AUXILIARY POLAR SYSTEMS=',I4,
      3     /10X,'VARIABLE ORDER ELEMENT OPTION=',I3,
      4     /10X,'TEMPERATURE OPTION=',I3,
      5     /10X,'THICKNESS OPTION=',I3,
      6     /10X,'RENUMBER OPTION=',I3/)
      WRITE (IO,150)
150   FORMAT(//10X,'AUXILIARY POLAR COORDINATE SYSTEMS')
      WRITE(IO,160)
160   FORMAT(/'SYSTEM NUMBER       X OFFSET          Y OFFSET')
      DO 170 I=1,IPOL
   READ(IN,*)  IPOLAR,XOFSET(IPOLAR),YOFSET(IPOLAR)
170   WRITE(IO,180)   IPOLAR,XOFSET(IPOLAR),YOFSET(IPOLAR)
180   FORMAT(//5X,I5,8X,F10.5,5X,F10.5)
185   DO 200 J=1,INBP
      READ(IN,*)  IPOLAR,K,XP(K,1),YP(K,1)
```

```
          IF(IPOLAR.EQ.0) GOTO 200
          RAD=XP(K,1)
          ANG=YP(K,1)*3.141592654/180.
          XP(K,1)=RAD*COS(ANG)+XOFSET(IPOLAR)
          YP(K,1)=RAD*SIN(ANG)+YOFSET(IPOLAR)
 200      CONTINUE
          IF(ITMP.EQ.0) GOTO 230
          DO 210 J=1,INBP

C++++++++++++++++++++++++++++++++
 210      READ(IN,*)  K,XP(K,2)
 220      FORMAT(I5,5X,F10.5)
C++++++++++++++++++++++++++++++++

 230  CONTINUE
          IF(ITHK.EQ.0) GOTO 250
          DO 240 J=1,INBP
 240      READ(IN,*)  K,YP(K,2)
 250      CONTINUE
          WRITE(IO,270)TITLE
 270      FORMAT(///1X,'PROBLEM:',20A4//)
          WRITE(IO,280)
 280  FORMAT(1X,'INPUT MESH COORDINATES'//1X,
    1'NUMBER      X-COORD           Y-COORD           TEMPERATURE   THICKNESS')
          DO 290 J=1,INBP

C++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 290      WRITE(IO,300) J,XP(J,1),YP(J,1),XP(J,2),YP(J,2)
 300      FORMAT(2X,I3,4(5X,F10.5))
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++

C-----------------------------------------------
C         LOOP ON THE REGIONS TO GENERATE THE ELEMENTS
C-----------------------------------------------
          MITY=0
          DO 340 J=1,INRG
          READ(IN,*)NRG,NROW1(NRG),NCOL1(NRG),IND1,(NOP(NRG,K),K=1,8,IND1),

          1NMAT(NRG),ITYPE(NRG)

          ITYPEE=ITYPE(NRG)

          IF(MITY.GE.ITYPEE) GOTO 340

          MITY=ITYPEE

340       IND11(NRG)=IND1
          IF(IVOE.EQ.0.AND.MITY.LE.2) MAXCON=8
C-----------------------------------------
          CALL REORDR(NOP,INRG,ICROSS,NCON)
C-----------------------------------------
C         REORDER THE INPUT REGIONS
          DO 360  J=1,INRG
          L=ICROSS(J)
          NDUMY(J,1)=NROW1(L)
          NDUMY(J,2)=NCOL1(L)
          NDUMY(J,3)=IND11(L)
          NDUMY(J,4)=NMAT(L)
          NDUMY(J,5)=ITYPE(L)
360   CONTINUE
          DO 370 J=1,INRG
          NROW1(J)=NDUMY(J,1)
          NCOL1(J)=NDUMY(J,2)
          IND11(J)=NDUMY(J,3)
          NMAT(J)=NDUMY(J,4)
          ITYPE(J)=NDUMY(J,5)
          DO 370 K=1,8
```

```
 370      NCON(J,K)=NOP(ICROSS(J),K)
          DO 380 J=1,INRG
          DO 380 K=1,8
          NOP(J,K)=NCON(J,K)
 380      CONTINUE
C-----------------------------------
C         MESHING INPUT SUPERELEMENTS
C-----------------------------------
          CALL MESH
C-----------------------------------
C         OUTPUT OF GENERATED INFORMATION
C-----------------------------------
          WRITE(IO,410) NB,NUMEL,IDIFF
 410      FORMAT(15X,'NB=',I4,5X,'NUMEL=',I4,5X,'IDIFF=',I3/)
          IF(IRENUM.EQ.1.AND.JRN.EQ.1) GOTO 430
          WRITE(IO,420)
 420      FORMAT(1H1///15X,'ORIGINAL NUMBERING SCHEME')
          GOTO 450
 430      WRITE(IO,440)
 440  FORMAT(1H1///15X,'OPTIMUM NUMBERING SCHEME')
 450      WRITE(IO,460)
 460      FORMAT(6X,'NODE',7X,'X-COORD',8X,'Y-COORD',4X,'TEMPERATURE',
    +4X,'THICKNESS'/)
          DO 470 J=1,NB
          K=J
          IF(IRENUM.EQ.1.AND.JRN.EQ.1) K=JMEM(J)
 470      WRITE(IO,480)  J,XNOD(K,1),YNOD(K,1),XNOD(K,2),YNOD(K,2)
 480      FORMAT(4X,I5,4(5X,F10.5))
          WRITE(IO,490)
 490      FORMAT(///5X,'ELEMENT',5X,'MATERIAL',10X,'NODES'/)
          DO 620 J=1,NUMEL
          NNODE=NCN(J)
          K=J
          IF(IRENUM.EQ.0.OR.JRN.EQ.0) GOTO 650
          DO 630 L=1,NNODE
          IL(L)=JNT(NCON(K,L))
 630  CONTINUE
          WRITE(IO,640)  K,MAT(K),(IL(L),L=1,NNODE)
          GOTO 620
 650      DO 660 L=1,NNODE
          IL(L)=NCON(K,L)
 660  CONTINUE
          WRITE(IO,640)  K,MAT(K),(IL(L),L=1,NNODE)
 640      FORMAT(5X,I5,5X,I5,5X,12I5)
 620  CONTINUE
          STOP
          END
c**********************************************************************
C
          SUBROUTINE MESH
C
c**********************************************************************
          COMMON XC(21,21),YC(21,21),TEMP(21,21),THK(21,21),NN(21,21),
          1NROWS,NCOL,IO,NCON(200,8),NUMEL,NB,NCN(200),IDIFF,IRENUM,JRN,
          2NM,NSE,MAXCON,MEM,NJT,MAT(200),NROW1(20),NCOL1(20),IND11(20),
          3NMAT(20),NOP(20,8),XP(100,2),YP(100,2),XNOD(441,2),YNOD(441,2),
          4INRG,ITYPE(20),JNT(441),MEMJT(10584),JMEM(441),NEWJT(441),
          5JOINT(441),       INBP,ITMP,ITHK

          COMMON /TM/JTE(200),ITY(200),IVOE
          DIMENSION TMPRGN(8),THKRGN(8)

          DIMENSION XRG(8),YRG(8),SF(8),NDN(8)

          DIMENSION NNRB(20,4,21),JT1(20,4)
          DIMENSION ICOMP(4,4),JTIK(20),NCHE(8)
          DATA ICOMP/-1,1,1,-1,1,-1,-1,1,1,-1,-1,1,-1,1,1,-1/
```

```
C------------------------------
C          INITIATION OF VARIABLES
C------------------------------
           IDIFF=0
           NB=0
           NEL=0
           DO 10 I=1,NSE
           DO 10 J=1,4
 10        JT1(I,J)=0
           DO 20 I=1,NM
           DO 20 J=1,NM
           XC(I,J)=0.0
           YC(I,J)=0.0
           THK(I,J)=0.0
           TEMP(I,J)=0.0
 20  NN(I,J)=0
C-------------------------------------
C          FINDING INPUT CONNECTIVITY DATA
C-------------------------------------
           DO 22 I=1,INRG
 22        JTIK(I)=0

           DO 33 I1=1,INRG
           DO 32 J1=1,4
           I11=(J1-1)*2+1
           N1=NOP(I1,I11)

           IF(IND11(I1).EQ.1) N2=NOP(I1,I11+1)
           IF(I11.EQ.7) I11=-1
           N3=NOP(I1,I11+2)

           DO 31 K1=1,INRG
           IF(K1.EQ.I1) GOTO 31

           DO 30 L1=1,4
           L11=(L1-1)*2+1
           NC1=NOP(K1,L11)

           IF(IND11(K1).EQ.1) NC2=NOP(K1,L11+1)

           IF(L11.EQ.7) L11=-1

           NC3=NOP(K1,L11+2)

           IF(IND11(K1).EQ.1.AND.IND11(I1).EQ.1) GOTO 25

           IF(N1.NE.NC3.OR.N3.NE.NC1) GOTO 30

           JT1(I1,J1)=K1
           GOTO 27

25         IF(N1.NE.NC3.OR.N2.NE.NC2.OR.N3.NE.NC1) GOTO 30
           JT1(I1,J1)=K1
27         IF(ITYPE(I1).LE.2.AND.ITYPE(K1).LE.2.OR.ITYPE(I1).GE.3.AND.
           1ITYPE(K1).GE.3) GOTO 30

           JTIK(I1)=J1
           JTIK(K1)=L1
30         CONTINUE

c++++++++++++++++++++++++++++++++++++++++++++++
c++++++++++++++++++++++++++++++++++++++++++++++

           WRITE (IO,40)
 40        FORMAT(//1X,'INPUT CONNECTIVITY DATA'/
           11X,'REGION      SIDE      1          2          3          4'/)
           DO 50 I=1,INRG
```

```
 50       WRITE(IO,60) I,(JT1(I,J),J=1,4)
 60       FORMAT(2X,I4,15X,4(I4,6X))
C         WRITING INPUT REGION INFORMATION
    WRITE(IO,70)
 70       FORMAT(/'INPUT REGION INFORMATION')
          WRITE(IO,80)
 80       FORMAT(/'REGION ROWS COLUMNS MATERIAL',7X,'BOUNDARY NODES'/)
          DO 410 KK=1,INRG
          NRG=KK
          NROWS=NROW1(KK)
          NCOL=NCOL1(KK)
          NMATTY=NMAT(KK)
          ITYO=ITYPE(KK)
          JTIKE=JTIK(KK)
          DO 90 I1=1,8
 90       NDN(I1)=NOP(KK,I1)
          IF(IND11(KK).EQ.2)NSNCN=4
          IF(IND11(KK).EQ.1)NSNCN=8

          IF(ITYO.EQ.1.OR.ITYO.EQ.2) GOTO 100
          GOTO 110
 100      NROWS=NROWS+1
          NCOL=NCOL+1
          GOTO 120
 110      NROWS=2*NROWS+1
          NCOL=2*NCOL+1
 120      CONTINUE
 130      FORMAT(1X,I3,4X,I3,4X,I3,4X,I4,4X,8I5)
          IF(NSNCN.EQ.4)GOTO 140
          WRITE(IO,130)NRG,NROW1(NRG),NCOL1(NRG),NMATTY,(NDN(I),I=1,8)
          GOTO 150
 140      CONTINUE
          WRITE(IO,130)NRG,NROW1(NRG),NCOL1(NRG),NMATTY,(NDN(ID),ID=1,7,2)
 150      CONTINUE
          IF(NROWS.GT.NM.OR.NCOL.GT.NM)   WRITE(IO,160)
 160      FORMAT(//5X,'NUMBER OF SUBDIVISIONS IN A SUPERELEMENT EXCEEDS',
          1          5X,'ALLOWABLE PROGRAM TERMINATED')
          IF(NROWS.GT.NM.OR.NCOL.GT.NM) STOP
C-----------------------------------------------
C         GENERATION OF THE ELEMENT NODAL COORDINATES
C-----------------------------------------------
          DO 200 I=1,8
          IF(NSNCN.NE.4) GOTO 190
          IF(I.EQ.1.OR.I.EQ.3.OR.I.EQ.5.OR.I.EQ.7) GOTO 190
          IS=NDN(I-1)
          IF(I.EQ.8) GOTO 170
          IE=NDN(I+1)
          GOTO 180
 170      IE=NDN(1)
 180      CONTINUE
          TMPRGN(I)=(XP(IS,2)+XP(IE,2))/2.
          THKRGN(I)=(YP(IS,2)+YP(IE,2))/2.
          XRG(I)=(XP(IS,1)+XP(IE,1))/2.
          YRG(I)=(YP(IS,1)+YP(IE,1))/2.
          GOTO 200
 190      CONTINUE
          II=NDN(I)
          TMPRGN(I)=XP(II,2)
          THKRGN(I)=YP(II,2)
          XRG(I)=XP(II,1)
          YRG(I)=YP(II,1)
 200      CONTINUE
          TR=NROWS-1
          DETA=2./TR
          TR=NCOL-1
          DSI=2./TR
C-----------------------------------------
```

```
C          GENERATION OF THE REGION NODE NUMBERS
C-------------------------------------------
           KN1=1
           KS1=1
           KN2=NROWS
           KS2=NCOL
           DO 280 I=1,4
           NRT=JT1(NRG,I)
           IF(NRT.EQ.NRG)   WRITE(IO,210)
 210  FORMAT(15X,'INPUT ERROR IN CONNECTEVITY,PROGRAM TERMINATED')
           IF(NRT.EQ.NRG) STOP
           IF(NRT.EQ.0.OR.NRT.GT.NRG)GOTO 280
           IF(ITYPE(NRG).NE.ITYPE(NRT)) GOTO 280
           I11=(I-1)*2+1
           N1=NOP(NRG,I11)
           IF(IND11(NRG).EQ.1) N2=NOP(NRG,I11+1)
           IF(I11.EQ.7) I11=-1
           N3=NOP(NRG,I11+2)
           DO 220 J=1,4
           L11=(J-1)*2+1
           NC1=NOP(NRT,L11)
           IF(IND11(NRT).EQ.1) NC2=NOP(NRT,L11+1)
           IF(L11.EQ.7) L11=-1
           NC3=NOP(NRT,L11+2)
           IF(IND11(NRG).EQ.1.AND.IND11(NRT).EQ.1) GOTO 215
           IF(N1.EQ.NC3.AND.N3.EQ.NC1) NRTS=J
           GOTO 220
 215       IF(N1.EQ.NC3.AND.N2.EQ.NC2.AND.N3.EQ.NC1) NRTS=J
 220  CONTINUE
           K=NCOL
           IF(I.EQ.2.OR.I.EQ.4) K=NROWS
           JL=1
           JK=ICOMP(I,NRTS)

           IF(JK.EQ.-1) JL=K
           DO 270 J=1,K
           GOTO (230,240,250,260),I
 230       NN(NROWS,J)=NNRB(NRT,NRTS,JL)
           KN2=NROWS-1
           GOTO 270
 240       NN(J,NCOL)=NNRB(NRT,NRTS,JL)
           KS2=NCOL-1
           GOTO 270
 250       NN(1,J)=NNRB(NRT,NRTS,JL)
           KN1=2
           GOTO 270
 260       NN(J,1)=NNRB(NRT,NRTS,JL)
           KS1=2
 270       JL=JL+JK
 280  CONTINUE
C-------------------------------------------------
           CALL CORD(NROWS,NCOL,DETA,DSI,XC,YC,XRG,YRG)
C-------------------------------------------------
           IF(KN1.GT.KN2) GOTO 365
           IF(KS1.GT.KS2) GOTO 365
           DO 360 I=KN1,KN2
           TR=I-1
           ETA=1.-TR*DETA
           DO 360 J=KS1,KS2
           IF(ITYO.NE.5) GOTO 290

           C1=MOD(I,2)
           C2=MOD(J,2)

           IF(C1.NE.0.0.OR.C2.NE.0.0) GOTO 290
           NN(I,J)=0
           GOTO 360
```

```
 290      TR=J-1
          SI=-1.+TR*DSI

          SF(1)=-.25*(1.-SI)*(1.-ETA)*(SI+ETA+1.)
          SF(2)=.50*(1.-SI**2)*(1.-ETA)
          SF(3)=.25*(1.+SI)*(1.-ETA)*(SI-ETA-1.)
          SF(4)=.50*(1.+SI)*(1-ETA**2)
          SF(5)=.25*(1.+SI)*(1.+ETA)*(SI+ETA-1.)
          SF(6)=.50*(1.-SI**2)*(1.+ETA)
          SF(7)=.25*(1.-SI)*(1.+ETA)*(ETA-SI-1.)
          SF(8)=.50*(1.-SI)*(1.-ETA**2)

          XC(I,J)=0.0
          YC(I,J)=0.0
          TEMP(I,J)=0.0
          THK(I,J)=0.0
          DO 300 K=1,8

C+++++++++++++++++++++++++++++++++++++
          XC(I,J)=XC(I,J)+XRG(K)*SF(K)
 300      YC(I,J)=YC(I,J)+YRG(K)*SF(K)
C+++++++++++++++++++++++++++++++++++++


          IF(ITMP.EQ.0) GOTO 320
          DO 310 K=1,8
 310      TEMP(I,J)=TEMP(I,J)+TMPRGN(K)*SF(K)
 320      IF(ITHK.EQ.0) GOTO 340
          DO 330 K=1,8
 330      THK(I,J)=THK(I,J)+THKRGN(K)*SF(K)
 340  CONTINUE
          NB=NB+1
          IF(NB.GT.NJT) WRITE(IO,350)
 350  FORMAT(15X,'NUMBER OF GENERATED NODES EXCEEDS ALLOWABLE',
          1/15X,'PRAGROM TERMINATED')
          IF(NB.GT.NJT) STOP

          X1=XC(I,J)
          Y1=YC(I,J)

C+++++++++++++++++++++++++++++++++++++++++++++++++
C         ELIMINATING DUPLICATE NODES AND PARAMETERS
C+++++++++++++++++++++++++++++++++++++++++++++++++

          ND=NB-1
          DO 701 NA=1,ND
          DX=ABS(X1-XNOD(NA,1))
          DY=ABS(Y1-YNOD(NA,1))
          IF(DX.GT.1.E-6.OR.DY.GT.1.E-6) GOTO 701
          YNOD(NA,2)=THK(I,J)
          XNOD(NA,2)=TEMP(I,J)

          XNOD(NA,1)=XC(I,J)
          YNOD(NA,1)=YC(I,J)

          NN(I,J)=NA
          NB=NB-1
          GOTO 360
 701      CONTINUE
          YNOD(NB,2)=THK(I,J)


          XNOD(NB,2)=TEMP(I,J)

          XNOD(NB,1)=XC(I,J)
          YNOD(NB,1)=YC(I,J)
```

```
          NN(I,J)=NB
 360  CONTINUE

C+++++++++++++++++++++++++++++++++++++++++
C         STORAGE OF THE BOUNDARY NODE NUMBERS
C+++++++++++++++++++++++++++++++++++++++++

 365      DO 370 I=1,NCOL
          NNRB(NRG,1,I)=NN(NROWS,I)
 370      NNRB(NRG,3,I)=NN(1,I)
          DO 380 I=1,NROWS
          NNRB(NRG,2,I)=NN(I,NCOL)
 380      NNRB(NRG,4,I)=NN(I,1)


          IF(ITYO.EQ.1.OR.ITYO.EQ.2) CALL LINEAR(NEL,NMATTY,ITYO,
    +JTIKE)
          IF(ITYO.EQ.3.OR.ITYO.EQ.4.OR.ITYO.EQ.5) CALL QUAD(NEL,
          1NMATTY,ITYO,JTIKE)


          NUMEL=NEL
 410  CONTINUE
          DO 420 J=1,NUMEL
          ITYE=ITY(J)
          IF(ITYE.EQ.1) NCN(J)=3
          IF(ITYE.EQ.2) NCN(J)=4
          IF(ITYE.EQ.3.OR.ITYE.EQ.4) NCN(J)=6
          IF(ITYE.EQ.5) NCN(J)=8
          IF (ITYE.NE.4) GOTO 420
          XNOD(NCON(J,4),1)=(XNOD(NCON(J,1),1)+XNOD(NCON(J,2),1))/2.
          YNOD(NCON(J,4),1)=(YNOD(NCON(J,1),1)+YNOD(NCON(J,2),1))/2.
          XNOD(NCON(J,5),1)=(XNOD(NCON(J,2),1)+XNOD(NCON(J,3),1))/2.
          YNOD(NCON(J,5),1)=(YNOD(NCON(J,2),1)+YNOD(NCON(J,3),1))/2.
          XNOD(NCON(J,6),1)=(XNOD(NCON(J,3),1)+XNOD(NCON(J,1),1))/2.
          YNOD(NCON(J,6),1)=(YNOD(NCON(J,3),1)+YNOD(NCON(J,1),1))/2.
 420  CONTINUE
          IF(IVOE.EQ.0) GOTO 430
C-------------------------------------------------------------------
C         FINDING COMMON BOUNDARIES BETWEEN LOWER AND HIGHER ORDER ELEMENTS
C-------------------------------------------------------------------

          MAE2=0
          DO 840 M1=1,NUMEL
          JTEE1=JTE(M1)
          ITYE1=ITY(M1)
          IF(JTEE1.EQ.0.OR.ITYE1.LE.2) GOTO 840
          KN1=3
          IF(ITYE1.EQ.5) KN1=4
          DO 830 L1=1,KN1
          I11=(L1-1)*2+1
          N1=NCON(M1,I11)
          N2=NCON(M1,I11+1)
          IF(L1.EQ.KN1) I11=-1
          N3=NCON(M1,I11+2)
          DO 820 M2=1,NUMEL
          IF(M2.EQ.M1.OR.M2.EQ.MAE2) GOTO 820
          JTEE2=JTE(M2)
          ITYE2=ITY(M2)
          IF(JTEE2.EQ.0.OR.ITYE2.GE.3) GOTO 820
          KN2=3
          IF(ITYE2.EQ.2) KN2=4
          DO 810 L2=1,KN2
          I22=L2
          NC1=NCON(M2,I22)
          IF(L2.EQ.KN2) I22=0
          NC3=NCON(M2,I22+1)
```

```
            IF(N1.NE.NC3.OR.N3.NE.NC1) GOTO 810
C-----------------------------------------------------------
C           INCREASING NODE ON THE BOUNDARY OF LOWER ORDER ELEMENT
C-----------------------------------------------------------
            MAE2=M2
            NCN(MAE2)=NCN(MAE2)+1
C-----------------------------------------------------
C           REARRAGEMENT NODES IN THE INCREASING NODE ELEMENT
C-----------------------------------------------------
            NO3=NCN(MAE2)
            DO 802 M3=1,NO3
 802        NCHE(M3)=NCON(MAE2,M3)
            IF(L2.EQ.KN2) GOTO 808
            DO 804 L3=L2+1,KN2
 804        NCON(MAE2,L3+1)=NCHE(L3)
 808        NCON(MAE2,L2+1)=N2
            GOTO 830
 810  CONTINUE
 820  CONTINUE
 830  CONTINUE
            MAE2=0
 840  CONTINUE


C**************** YAOWU Liu and KUAN CHEN  1044 ******************


 430        IF(IRENUM.EQ.0) GOTO 440

C           RENUMBERING NODES FOR REDUCING BANDWIDTH

C--------------------
            CALL RENMBR
C--------------------
 440  CONTINUE
            RETURN
            END
C*****************************************************************
C
            SUBROUTINE LINEAR(NEL,NMATTY,ITYO,JTIKE)
C
C*****************************************************************
            COMMON XC(21,21),YC(21,21),TEMP(21,21),THK(21,21),NN(21,21),
            1NROWS,NCOL,IO,NCON(200,8),NUMEL,NB,NCN(200),IDIFF,IRENUM,JRN,
            2NM,NSE,MAXCON,MEM,NJT,MAT(200),NROW1(20),NCOL1(20),IND11(20),
            3NMAT(20),NOP(20,8),XP(100,2),YP(100,2),XNOD(441,2),YNOD(441,2),
            4INRG,ITYPE(20),JNT(441),MEMJT(10584),JMM1(441),NEWJT(441),
            5JOINT(441),INBP,ITMP,ITHK

            COMMON /TM/JTE(200),ITY(200),IVOE
            DIMENSION NE(441),NR(9),LB(3),LC(4)
            K=1
            DO 10 I=1,NROWS
            DO 10 J=1,NCOL
            NE(K)=NN(I,J)
 10         K=K+1
            L=NROWS-1
            DO 90 I=1,L
            DO 90 J=2,NCOL
            NR(1)=NCOL*I+J-1
            NR(2)=NCOL*I+J
            NR(3)=NCOL*(I-1)+J
            NR(4)=NCOL*(I-1)+J-1
            IF(ITYO.EQ.1) GOTO 20
            NEL=NEL+1
            JTE(NEL)=0
            ITY(NEL)=ITYO
```

```
      IF(JTIKE.EQ.0) GOTO 15
      IF(JTIKE.EQ.1.AND.I.EQ.L)     JTE(NEL)=1
      IF(JTIKE.EQ.2.AND.J.EQ.NCOL)  JTE(NEL)=2
      IF(JTIKE.EQ.3.AND.I.EQ.1)     JTE(NEL)=3
      IF(JTIKE.EQ.4.AND.J.EQ.2)     JTE(NEL)=4
15    IF(NEL.GT.MEM)GOTO 70
      GOTO 60
20    DIAG1=SQRT((XC(I,J)-XC(I+1,J-1))**2+(YC(I,J)-YC(I+1,J-1))**2)
      DIAG2=SQRT((XC(I+1,J)-XC(I,J-1))**2+(YC(I+1,J)-YC(I,J-1))**2)
      DO 50 IJ=1,2
      NEL=NEL+1
      JTE(NEL)=0
      ITY(NEL)=ITYO
      IF(JTIKE.EQ.0) GOTO 25
      IF(JTIKE.EQ.1.AND.I.EQ.L)     JTE(NEL)=1
      IF(JTIKE.EQ.2.AND.J.EQ.NCOL)  JTE(NEL)=2
      IF(JTIKE.EQ.3.AND.I.EQ.1)     JTE(NEL)=3
      IF(JTIKE.EQ.4.AND.J.EQ.2)     JTE(NEL)=4
25    IF(NEL.GT.MEM)GOTO 70
      IF((DIAG1/DIAG2).GT.1.02) GOTO 30
      J1=NR(1)
      J2=NR(IJ+1)
      J3=NR(IJ+2)
      GOTO 40
30    J1=NR(IJ)
      J2=NR(IJ+1)
      J3=NR(4)
40    CONTINUE
      NCON(NEL,1)=NE(J1)
      NCON(NEL,2)=NE(J2)
      NCON(NEL,3)=NE(J3)
      MAT(NEL)=NMATTY
      ITY(NEL)=ITYO
      DO 41 K1=1,3
41    LB(K1)=NCON(NEL,K1)
      DO 42 K1=1,2
      IK=K1+1
      DO 42 K2=IK,3
      NSX=IABS(LB(K1)-LB(K2))
      IF(NSX.LE.IDIFF) GOTO 42

      IDIFF=NSX
42    CONTINUE
50    CONTINUE
      GOTO 90
60    J1=NR(1)
      J2=NR(2)
      J3=NR(3)
      J4=NR(4)
      NCON(NEL,1)=NE(J1)
      NCON(NEL,2)=NE(J2)
      NCON(NEL,3)=NE(J3)
      NCON(NEL,4)=NE(J4)
      MAT(NEL)=NMATTY
      ITY(NEL)=ITYO
      DO 61 K1=1,4
61    LC(K1)=NCON(NEL,K1)
      DO 62 K1=1,3
      IK=K1+1
      DO 62 K2=IK,4
      NSX=IABS(LC(K1)-LC(K2))
      IF(NSX.LE.IDIFF) GOTO 62
      IDIFF=NSX
62    CONTINUE
      GOTO 90
70    WRITE(IO,80)
80    FORMAT(15X,'NUMBER OF GENERATED ELEMENTS EXCEEDS ALLOWABLE',
```

```
      1 /15X,'PROGRAM TERMINATED')
      STOP
 90   CONTINUE
      RETURN
      END
c*****************************************************************
C
      SUBROUTINE QUAD(NEL,NMATTY,ITYO,JTIKE)
C
c*****************************************************************
      COMMON XC(21,21),YC(21,21),TEN2(21,21),THK(21,21),NN(21,21),
      1NROWS,NCOL,IO,NCON(200,8),NUMEL,NB,NCN(200),IDIFF,IRENUM,JRN,
      2NM,NSE,MAXCON,MEM,NJT,MAT(200),NROW1(20),NCOL1(20),IND11(20),
      3NMAT(20),NOP(20,8),XP(100,2),YP(100,2),XNOD(441,2),YNOD(441,2),
      4INRG,ITYPE(20),JNT(441),MEMJT(10584),JMEM(441),NEWJT(441),
      5JOINT(441),INBP,ITMP,ITHK

      COMMON /TM/JTE(200),ITY(200),IVOE
      DIMENSION NE(441),NR(9),LB(6),LC(8)
      K=1
      DO 10 I=1,NROWS
      DO 10 J=1,NCOL
      NE(K)=NN(I,J)
 10   K=K+1
      L=(NROWS-1)/2
      LL=NCOL-1

      DO 90 I=1,L
      DO 90 J=2,LL,2

      NR(1)=2*NCOL*I+J-1
      NR(2)=2*NCOL*I+J
      NR(3)=2*NCOL*I+J+1
      NR(4)=NCOL*(2*I-1)+J+1
      NR(5)=NCOL*(2*I-1)-(NCOL-1)+J
      NR(6)=NCOL*(2*I-1)-(NCOL-1)+J-1
      NR(7)=NCOL*(2*I-1)-(NCOL-1)+J-2
      NR(8)=NCOL*(2*I-1)+J-1
      NR(9)=NCOL*(2*I-1)+J

      IF(ITYO.EQ.3.OR.ITYO.EQ.4) GOTO 20
      NEL=NEL+1
      JTE(NEL)=0
      ITY(NEL)=ITYO
      IF(JTIKE.EQ.0) GOTO 15
      IF(JTIKE.EQ.1.AND.I.EQ.L)   JTE(NEL)=1
      IF(JTIKE.EQ.2.AND.J.EQ.LL)  JTE(NEL)=2
      IF(JTIKE.EQ.3.AND.I.EQ.1)   JTE(NEL)=3
      IF(JTIKE.EQ.4.AND.J.EQ.2)   JTE(NEL)=4
 15   IF(NEL.GT.MEM) GOTO 70
      GOTO 60
 20   II=(2*I-1)
      DIAG1=SQRT((XC(II,J+1)-XC(II+2,J-1))**2+(YC(II,J+1)-YC(II+2,J-1))
      1**2)
      DIAG2=SQRT((XC(II+2,J+1)-XC(II,J-1))**2+(YC(II+2,J+1)-YC(II,J-1))
      1**2)
      FR=1
      IL=1
      DO 50 IJ=1,2
      NEL=NEL+1

      JTE(NEL)=0
      ITY(NEL)=ITYO

      IF(JTIKE.EQ.0)     GOTO 25
 25   IF(NEL.GT.MEM) GOTO 70
```

```
      IF((DIAG1/DIAG2).GT.1.02)      GOTO 30

      J1=NR(IL)
      J2=NR(IL+1)
      J3=NR(IL+2)
      J4=NR(IL+3)

      J5=NR(FR+4)

      J6=NR(9)

C     J1=NR(1)
C     J2=NR(2)
C     J3=NR(3)
C     J4=NR(4)
C     J5=NR(5)
C     J6=NR(9)

C     J1=NR(5)
C     J2=NR(6)
C     J3=NR(7)
C     J4=NR(8)
C     J5=NR(1)
C     J6=NR(9)
      GOTO 40
30    J1=NR(IK+6)
      J2=NR(IL)
      J3=NR(IL+2)
      J4=NR(IK+7)
      J5=NR(IL+1)
      J6=NR(9)
40    CONTINUE
      IL=IL+4
      FR=FR-4
      NCON(NEL,1)=NE(J1)
      NCON(NEL,2)=NE(J2)
      NCON(NEL,3)=NE(J3)
      NCON(NEL,4)=NE(J4)
      NCON(NEL,5)=NE(J5)
      NCON(NEL,6)=NE(J6)
      MAT(NEL)=NMATTY
      DO 41 K1=1,6
41    LB(K1)=NCON(NEL,K1)
      DO 42 K1=1,5
      IK=K1+1
      DO 42 K2=IK,6
      NSX=IABS(LB(K1)-LB(K2))
      IF(NSX.LE.IDIFF) GOTO 42
      IDIFF=NSX
42    CONTINUE
50    CONTINUE
      GOTO 90
60    J1=NR(1)
      J2=NR(2)
      J3=NR(3)
      J4=NR(4)
      J5=NR(5)
      J6=NR(6)
      J7=NR(7)
      J8=NR(8)
      NCON(NEL,1)=NE(J1)
      NCON(NEL,2)=NE(J2)
      NCON(NEL,3)=NE(J3)
      NCON(NEL,4)=NE(J4)
      NCON(NEL,5)=NE(J5)
      NCON(NEL,6)=NE(J6)
      NCON(NEL,7)=NE(J7)
```

```
          NCON(NEL,8)=NE(J8)
          MAT(NEL)=NMATTY
          DO 61 K1=1,8
61        LC(K1)=NCON(NEL,K1)
          DO 62 K1=1,7
          IK=K1+1
          DO 62 K2=IK,8
          NSX=IABS(LC(K1)-LC(K2))
          IF(NSX.LE.IDIFF) GOTO 62
          IDIFF=NSX
62        CONTINUE
          GOTO 90
70        WRITE(IO,80)
80        FORMAT(15X,'NUMBER OF GENERATED ELEMENTS EXCEEDS ALLOWBLE',
          1/15X,'PROGARM TERMINATED')
          STOP
 90       CONTINUE
          RETURN
          END
C*********************************************************************
C
          SUBROUTINE REORDR(NOP,INRG,ICROSS,NCON)
C
C*********************************************************************
          DIMENSION NOP(20,8),NCON(200,8),IUSED(20),ICROSS(20),NODE(4)
          DO 10 I=1,INRG
 10       IUSED(I)=1
          IUSED(1)=-1
          ICROSS(1)=1
          KT=2
          DO 30 I=1,INRG
          NODE(1)=NOP(ICROSS(I),1)
          NODE(2)=NOP(ICROSS(I),3)
          NODE(3)=NOP(ICROSS(I),5)
          NODE(4)=NOP(ICROSS(I),7)
          IC=ICROSS(I)
          DO 35 J=1,INRG
          IF(I.EQ.J) GOTO 35
          IF(IUSED(J).LT.0) GOTO 35
          M=0
          DO 20 K=1,8,2
          DO 20 L=1,4
 20       IF(NOP(J,K).EQ.NODE(L).AND.IUSED(J).GT.0) M=M+1
          IF(M.EQ.0.OR.J.EQ.IC) GOTO 35
          ICROSS(KT)=J
          KT=KT+1
          IUSED(J)=-1
 35       CONTINUE
 30       CONTINUE
          DO 50 I=1,INRG
          DO 50 J=1,8
 50       NCON(I,J)=NOP(ICROSS(I),J)
          RETURN
          END
C*********************************************************************
C
          SUBROUTINE CORD(NROWS,NCOL,DETA,DSI,XC,YC,XRG,YRG)
C
c*********************************************************************
          DIMENSION XC(21,21),YC(21,21),XRG(8),YRG(8),SF(8)
          DO 365 I=1,NROWS
          TR=I-1
          ETA=1.-TR*DETA
          DO 360 J=1,NCOL
          TR=J-1
          SI=-1.+TR*DSI
          SF(1)=-.25*(1.-SI)*(1.-ETA)*(SI+ETA+1.)
```

```
            SF(2)=.50*(1.-SI**2)*(1.-ETA)
            SF(3)=.25*(1.+SI)*(1.-ETA)*(SI-ETA-1.)
            SF(4)=.50*(1.+SI)*(1.-ETA**2)
            SF(5)=.25*(1.+SI)*(1.+ETA)*(SI+ETA-1.)
            SF(6)=.50*(1.-SI**2)*(1.+ETA)
            SF(7)=.25*(1.-SI)*(1.+ETA)*(ETA-SI-1.)
            SF(8)=.50*(1.-SI)*(1.-ETA**2)
            XC(I,J)=0.
            YC(I,J)=0.
            DO 300 K=1,8
C+++++++++++++++++++++++++++++++++++++++++
            XC(I,J)=XC(I,J)+XRG(K)*SF(K)
 300        YC(I,J)=YC(I,J)+YRG(K)*SF(K)
C+++++++++++++++++++++++++++++++++++++++++

 360  CONTINUE
 365  CONTINUE
            RETURN
            END
C*********************************************************************
C
            SUBROUTINE RENMBR
C
C*********************************************************************
            COMMON XC(21,21),YC(21,21),TEMP(21,21),THK(21,21),NN(21,21),
            1NROWS,NCOL,IO,NCON(200,8),NUMEL,NB,NCN(200),IDIFF,IRENUM,JRN,
            2NM,NSE,MAXCON,MEM,NJT,MAT(200),NROW1(20),NCOL1(20),IND11(20),
    3NMAT(20),NOP(20,8),XP(100,2),YP(100,2),XNOD(441,2),YNOD(441,2),
    4JOINT(441),ITYPE(20),JNT(441),MEMJT(10584),JMEM(441),NEWJT(441),
    5INRG,INBP,ITMP,ITHK

     EQUIVALENCE (NCON(1,1),JT(1)),(IW,IO)
     EQUIVALENCE (NB,NUMNP),(NEL,NUMEL)
     DIMENSION JT(1600)
     DO 10 I=1,NJT
     JNT(I)=0
     JMEM(I)=0
 10          CONTINUE
    IJ=NJT*MAXCON
     DO 20 I=1,IJ
 20          MEMJT(I)=0
    DO 70 J=1,NUMEL
     NNODE=NCN(J)
     DO 60 I=1,NNODE
     JNTI=JT(MEM*(I-1)+J)
     IF(JNTI.EQ.0) GOTO 70
     JSUB=(JNTI-1)*MAXCON
     DO 50 II=1,NNODE
     IF(II.EQ.I) GOTO 50
     JJT=JT(MEM*(II-1)+J)
     IF(JJT.EQ.0) GOTO 60
            MEM1=JMEM(JNTI)
            IF(MEM1.EQ.0) GOTO 40
            DO 30 III=1,MEM1
            IF(MEMJT(JSUB+III).EQ.JJT) GOTO 50
 30         CONTINUE
 40         JMEM(JNTI)=JMEM(JNTI)+1
            MEMJT(JSUB+JMEM(JNTI))=JJT
            IF(IABS(JNTI-JJT).GT.IDIFF)  IDIFF=IABS(JNTI-JJT)
 50         CONTINUE
 60         CONTINUE
 70         CONTINUE
            WRITE(IW,80) IDIFF
 80         FORMAT(//5X,'ORIGINAL NODE DIFFERENCE',I4//)
C-------------------------------------------------------------
            CALL OPTNUM
C-------------------------------------------------------------
```

```
 90         RETURN
            END
C*****************************************************************
C
            SUBROUTINE OPTNUM
C
C*****************************************************************
            COMMON XC(21,21),YC(21,21),TEMP(21,21),THK(21,21),NN(21,21),
           1NROWS,NCOL,IO,NCON(200,8),NUMEL,NB,NCN(200),IDIFF,IRENUM,JRN,
           2NM,NSE,MAXCON,MEM,NJT,MAT(200),NROW1(20),NCOL1(20),IND11(20),
           3NMAT(20),NOP(20,8),XP(100,2),YP(100,2),XNOD(441,2),YNOD(441,2),
           4INRG,ITYPE(20),JNT(441),MEMJT(10584),JMEM(441),NEWJT(441),
           5JOINT(441),INBP,ITMP,ITHK
            EQUIVALENCE (IW,IO)
            EQUIVALENCE (NB,NUMNP),(NEL,NUMEL)
            JRN=1
            MINMAX=IDIFF
            DO 70 IK=1,NUMNP
            DO 10 J=1,NUMNP
            JOINT(J)=0
 10         NEWJT(J)=0
            MAX=0
            I=1
            NEWJT(1)=IK
            JOINT(IK)=1
            K=1
 20         K4=JMEM(NEWJT(I))
            IF(K4.EQ.0) GOTO 40
            JSUB=(NEWJT(I)-1)*MAXCON
            DO 30 JJ=1,K4
            K5=MEMJT(JSUB+JJ)
            IF(JOINT(K5).GT.0) GOTO 30
            K=K+1
            NEWJT(K)=K5
            JOINT(K5)=K
            NDIFF=IABS(I-K)
            IF(NDIFF.GE.MINMAX) GOTO 70
            IF(NDIFF.GT.MAX) MAX=NDIFF
 30   CONTINUE
    IF(K.EQ.NUMNP) GOTO 50
 40   I=I+1
    GOTO 20
 50         MINMAX=MAX
    DO 60 J=1,NUMNP
 60   JNT(J)=JOINT(J)
 70   CONTINUE
    IF(MINMAX.LT.IDIFF) GOTO 90
    JRN=0
    WRITE (IW,80)
 80   FORMAT(/5X,'ORIGINAL NUMBERING SCHEME RETAINED'/
           15X,'SEMI-BANDWITH REMAINS THE SAME'/)
    GOTO 150
 90         IDIFF=MINMAX
    WRITE(IW,100) IDIFF
100  FORMAT(/5X,'NODE DIFFERENCE AFTER RENDMBERING=',I4//)
           DO 110 I=1,NUMNP
110  JMEM(JNT(I))=I
150  CONTINUE
    RETURN
    END
```

INTRODUCTION GÉNÉRALE

CHAPITRE 1

Etat de l'art

CHAPITRE 2

MÉTHODES DE GÉNÉRATION DE MAILLAGE

CHAPITRE 3

Méthodes de blocs

CHAPITRE 4

Présentation du programme développé

CHAPITRE 5

Application

CONCLUSION GÉNÉRALE

Références bibliographiques

Annexe 1

La Méthode des éléments finis

Annexe 2

Le fichier des données et des résultats

Annexe 3

Le programme