

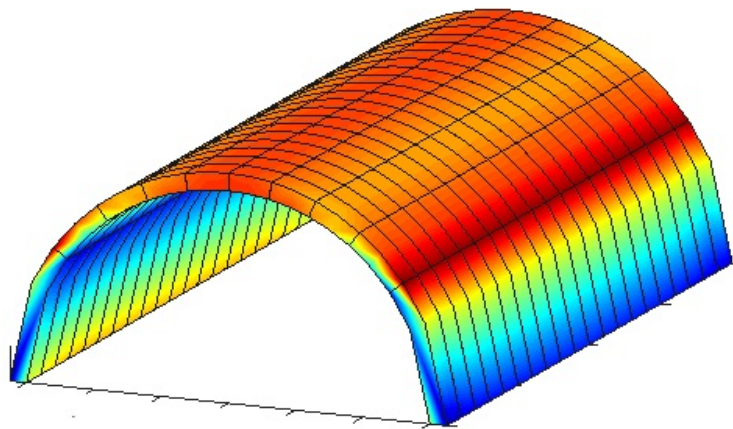
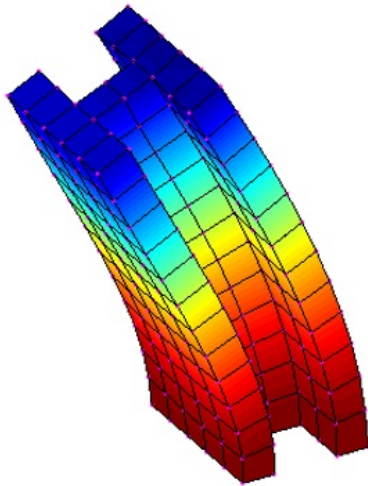
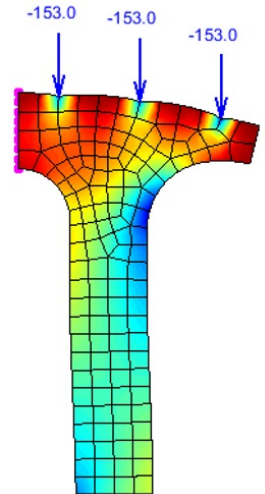


Mémoire

En vue de l'obtention du diplôme de mastère

Filière : Génie Civil

Option : Structure Académique



Thème :

**Elaboration d'un programme
d'éléments finis en statique et dynamique
pour l'analyse des structures 1D ,2D et 3D**

Apl_design MARS version 1.0

Présenté par :

M'RAIN SOHEIR

Encadré par :

Mr BENGRAH Reda

Remerciements

Voilà une bien belle aventure qui s'achève et que je regrette déjà. Elle aura été riche tant sur le plan professionnel que personnel.

Tous mes remerciements à ceux qui m'ont aidé tout au long de mon parcours en master II à savoir tous mes enseignants du département de Génie Civil université de Boumerdes UMBB

Dédicaces

A mes parents

A mon mari

A mes deux Filles

Résumé

La méthode des éléments finis représente l'une des réalisations les plus significatives dans le domaine des méthodes de calcul où l'implémentation informatique a permis le passage de la théorie à la formulation en éléments finis tout en s'appuyant sur le bagage mathématique.

Dans ce projet de fin d'étude il a été élaboré un programme d'éléments finis sous l'environnement **Matlab** sous le nom d'**Apl Design Structure MARS** version 1.0 où plusieurs type d'éléments finis ont été formulés en déplacement pour l'analyse des structures (**treillis, poutres, structures planes et structures 3D**) et cela dans le domaine **élastique linéaire, statique et dynamique**.

Dans ce travail un grand intérêt a été porté au côté interactif du programme :

- Un pré-traitement : génération de maillage automatique, visualisation de la structure, affichage des numéros de nœuds, numéros d'éléments et des conditions au limites.
- Possibilité d'accéder au maximum d'informations ou données pendant la résolution des problèmes
 - Visualisation en surbrillance de n'importe quel élément fini avec une couleur différente de celle de la structure.
 - Visualisation de la forme de la matrice de rigidité globale et sa demie largeur de bande.
 - L'affichage de la position de la matrice de rigidité de l'élément au choix, dans la matrice de rigidité globale.
 - L'affichage ou l'impression de la matrice de rigidité de n'importe quel élément finis.
 - Afficher l'élément fini avec son numéro, les numéros de nœuds, les positions et les numéros des points de gauss ainsi que les contraintes et contraintes principales correspondantes.
- Un post-traitement (aperçus et simulation de la structure déformée, diagramme des contraintes, comportement de la structure sous effet dynamique etc...).

MOTS CLÉS: MEF, Matlab Méthode des éléments finis, Analyse linéaire, Statique, Dynamique

ABSTRACT

The finite element method represents one of the most significant achievements in the field of computational methods where the computer implementation has allowed the transition from theory to finite element formulation while relying on the mathematical background.

In this end of project study a finite element program was developed under the Matlab environment under the name of **Apl Design Structure MARS** version 1.0 where several types of finite elements were formulated in displacement for the analysis of structures. (truss, beams, plane structures and 3D structures) and that in the linear elastic domain; static and dynamic.

In this work a great interest was brought to the interactive side of the program:

→ A preprocessing: automatic mesh generation, visualization of the structure, display of node numbers, element numbers and boundary conditions.

→ the academic side has been thought of which allows to access as much information or data as possible during problem solving so the details that it is possible to see are:

- Highlighted visualization of any finished element with a color different from that of the structure.
- Visualization of the shape of the global stiffness matrix and its half band width.
- The display of the position of the stiffness matrix of the element of your choice, in the global stiffness matrix.
- Display or printing of the stiffness matrix of any finite element.
- Display the finite element with its number, the node numbers, the positions and the numbers of the gauss points as well as the corresponding main constraints and constraints.

→ A postprocessing (insights and simulation of the deformed structure, stress diagram, behavior of the structure under dynamic effect etc...).

KEYWORDS: FEM, Matlab Finite element method, Linear analysis, Statics, Dynamics

تمثل طريقة العناصر المحدودة أحد أهم الإنجازات في مجال الأساليب الحسابية حيث أتاح تطبيق الكمبيوتر الانتقال من النظرية إلى صياغة العناصر المحدودة، مع الاعتماد على الخلفية الرياضية.

في نهاية مشروع الدراسة هذا، تم تطوير برنامج العناصر المحدودة في بيئة Matlab تحت اسم

Apl Design Structure MARS الإصدار 1.0

حيث تمت صياغة عدة أنواع من العناصر المحدودة في حالة الإزاحة لتحليل الهياكل. (اعمدة، عوارض، هياكل مستوية وهياكل ثلاثية الأبعاد) وذلك في المجال المرن الخطي؛ الثابت والديناميكي.

في هذا العمل، كان الاهتمام منصباً على الجانب التفاعلي في البرنامج:

معالجة مسبقة: إنشاء شبكة تلقائية، تصور الهيكل، عرض أرقام العقد، أرقام العناصر والشروط الحدودية.

تم التفكير في الجانب الأكاديمي الذي يسمح بالوصول إلى أكبر قدر ممكن من المعلومات أو البيانات أثناء حل المشكلات ومن بين التفاصيل التي يمكن رؤيتها

- إبراز استخراج أي عنصر محدد بلون مختلف عن لون الهيكل.
- تصور لشكل مصفوفة المرونة الكلية وعرض نصف عرض شريطها
- عرض موضع مصفوفة المرونة للعنصر الذي تختاره، في مصفوفة المرونة الإجمالية.
- عرض أو طباعة مصفوفة المرونة لأي عنصر محدد.
- عرض العنصر المحدود برقمه، وأرقام العقد، والمواضع وأرقام نقاط جاوس بالإضافة إلى الضغط والضغط الرئيسية الموافق

معالجة لاحقة (رؤيا ومحاكاة للبنية المشوهة، مخطط الإجهاد، سلوك الهيكل تحت التأثير الديناميكي، إلخ ...).

الكلمات الرئيسية: طريقة العناصر المحدودة، دراسة خطية الستاتيك و الديناميكا.

Table des matières

Remerciements	1
Résumé	3
ABSTRACT	4
ملخص	5
Table des matières	6
Notations	11
Abréviations	13
Introduction Générale	14
Contexte de l'étude	14
Cadre et objectif du projet de fin d'étude :	15
Organisation du manuscrit	16
Chapitre 1 Étude bibliographique	17
Sommaire	18
1. Introduction.....	19
1.1. Introduction.....	19
1.2. Equation de base de la théorie de l'élasticité linéaire.	19
1.2.1. Équations d'équilibre	19
1.2.2. Equations de compatibilité	20
1.2.3. Equations définissant la loi de comportement (06 eq)	20
1.3. Relation contraintes déformations :	21
1.3.1. Elasticité plane	21
1.3.1.1. Contrainte plane.....	22
1.3.1.2. Déformation plan	22
1.4. Méthodes de résolutions	23
1.4.1. Méthode des déplacements	23
1.4.2. Organigramme de calcul	24
1.4.3. Minimisation de l'énergie potentielle totale	25

Chapitre 2	Methodes des éléments finis	27
	Sommaire	28
2.1	Introduction.....	29
2.2	Discretisation du milieu.....	30
2.3	Types d'éléments finis.....	32
2.4	Fonctions de forme	35
2.5	Application du principe des travaux virtuels :.....	36
2.6	Choix des éléments programmés.....	37
2.7	Formulation des éléments finis.....	38
2.7.1	Interpolation de déplacement :	38
2.7.2	Interpolation de la déformation :.....	38
2.7.3	Interpolation de la contrainte :	38
2.7.4	Matrice de rigidité élémentaire :	39
2.8	Éléments de référence 1D.....	40
2.8.1	Élément barre.....	40
2.9	Éléments de référence 2D.....	41
2.9.1	Élément Poutre	41
2.10	Éléments iso-paramétriques	42
2.10.1	Élément triangulaire à 6 nœuds T6.....	42
2.10.2	Élément quadratique iso-paramétrique à huit nœuds	44
2.11	Éléments de référence 3D.....	47
2.11.1	Élément Tétraèdre a 4 nœuds.....	48
2.11.2	Élément hexaèdre à 8 nœuds	50
2.11.3	Élément hexaèdre a 20 nœuds	54
2.12	Vecteur forces	57
2.13	Conditions aux limites :	57
2.14	Méthode de factorisation de Cholesky	58
2.15	Intégration numérique	60
2.16	Les méthodes de résolution des équations d'équilibre	66
2.16.1	Méthode de superposition modale.....	66
2.16.2	Intégration pas à pas	69
2.16.3	Les méthodes d'intégrations directe.....	69
2.16.4	Method de Newmark	70
2.16.5	Algorithme de la méthode de Wilson θ	71

Chapitre 3	Présentation du Programme.....	73
	Sommaire	74
3	Introduction	76
3.1	Introduction.....	76
3.2	Présentation du langage de programmation Matlab.....	78
3.3	Structure d'un programme d'éléments finis	79
3.4	Structure du fichier de données.....	81
3.4.1	Fichier de données pour l'élément barre.....	82
3.4.2	Fichier de données pour l'éléments Poutre	83
3.4.3	Fichier de données pour les éléments 2D	84
3.4.4	Fichier de données pour les éléments 3D	85
3.5	Génération de maillage	86
3.6	Apl_Élément barre	95
3.6.1	Interface du programme	95
3.6.2	Sous programmes.....	99
3.6.3	Organigramme de résolution	100
3.7	Apl Elément Poutre	101
3.7.1	Interface du programme Apl_Poutre.....	101
3.7.2	Sous programmes.....	105
3.7.3	Organigramme de résolution et sousroutines.....	106
3.8	Apl Elément triangulaire T6.....	108
3.8.1	Interface du programme	108
3.8.2	Sous programmes.....	110
3.8.3	Organigramme de calculs.....	111
3.9	Apl Elément quadratique iso-paramétrique Q8.....	112
3.9.1	Interface du programme	112
3.9.2	Sous programmes.....	115
3.9.3	Organigramme de résolution	116
3.10	Apl Elément tétraédrique linéaire (T4)	117
3.10.1	Interface du programme	117
3.10.2	Sous programmes.....	119
3.10.3	Organigramme de résolution	120
3.11	Apl Elément brique linéaire (solide C8).....	121
3.11.1	Interface du programme	121
3.11.2	Sous Programmes.....	123
3.11.3	Organigramme de résolution	125

3.12	Apl Elément hexaédron (C20)	126
3.12.1	Interface du programme	126
3.12.2	Sous programme	128
3.12.3	Organigramme de résolution	129
Chapitre 4	Validation du Programme Apl_Design MARS_{Version1.0}	130
	Sommaire	131
4	Introduction	132
4.1	Introduction.....	132
4.2	Apl design Barre	133
4.3	Apl design Poutre	137
4.4	Apl design Q8	139
4.5	Apl design T6	147
4.6	Apl design Tetra4.....	153
4.7	Apl design Hexa20.....	158
4.8	Apl design Cub8.....	166
4.9	Conclusion	173
Chapitre 5	Calculs numériques.....	178
5	Introduction	179
5.1	Introduction.....	179
5.2	Apl design Q8	180
5.2.1	Exemple 01	180
5.2.2	Exemple 02	186
5.2.3	Exemple 03	190
5.2.4	Exemple 04	194
5.3	Apl design T6	199
5.3.1	Exemple 01	199
5.3.2	Exemple 02	204
5.4	Apl design Cub8.....	209
5.4.1	Exemple 01	209
5.4.2	Exemple 02	213
5.5	Apl design Hexa20.....	218
5.5.1	Exemple 01	218
5.5.2	Exemple 02	221
5.5.3	Exemple 03	227
5.6	Conclusion générale	232

Bibliographies	235
Annexe A.....	237
Sommaire	238
Apl_lect_dones.....	239
Apl_calcul_barre.....	240
Apl_calcul_poutres.....	245
Apl_Factor_Cholesky.....	250
Apl_newmark_implusion.....	251
Apl_M_rigidite_elementQ8.....	252
Apl_matrix_B.....	257
Apl_Fonct_Forme.....	257
Apl_M_rigidite_triglT6.....	258
Apl_Fct_Form_trigl.....	261
Apl_Tetra_free.....	262
Apl_matrice_D.....	265
Apl_Point_Gauss_tetra.....	266
Apl_BJdet_tetra.....	267
Cprinc1.....	267
Apl_cub_8N.....	268
Apl_Fonct_Forme_cub8.....	272
Apl_Cube_20N_free.....	273
Apl_Fonct_Forme_cub20.....	277
Apl_Points_Gauss.....	278
Annexe B.....	279
Table des figures	282
Liste des tableaux.....	287

Notations

$\{ \}$	Matrice colonne
$[]$	Matrice
b_i	Vecteur des forces de volume
f_i	Vecteur des forces de surface
$u_{,ij}$	Vecteur des déplacements
ϵ_{ij}	Tenseur des déformations
ϵ	Déformations réelles
σ	Contraintes
σ_{ij}	Tenseur des contraintes
λ, μ	Coefficients de Lamé
ν	Coefficient de Poisson
ρ	Masse volumique
E	Module d'élasticité
G	Module de cisaillement
δ_{ij}	Symbole de Kronecker
W_{ext}	Travail des forces extérieures
W_{int}	Travail interne
Π	Energie potentielle totale
T	Energie cinétique potentielle des forces appliquées
U	Energie de déformation
V	Energie potentielle totale
$\sigma_{xx} \sigma_{yy} \sigma_{zz}$	Contraintes normales
$\tau_{xy} \tau_{xz} \tau_{yz}$	Contraintes de cisaillements
t	Epaisseur
u	Déplacement suivant l'axe des x
v	Déplacement suivant l'axe des y
w	Déplacement suivant l'axe des z

u	Déplacement élémentaire
u_i, v_i, w_i	Déplacements nodaux dans les directions x, y et z
x, y, z	Coordonnées cartésiennes
ζ, η, ξ	Coordonnées isoparamétriques
[N]	Matrice des fonctions de forme
[D]	Matrice d'élasticité
[J]	Matrice Jacobienne
[KG]	Matrice de rigidité global
[B]	Matrice reliant les déformations aux variables nodales
[Ke]	Matrice de rigidité de l'élément
{Fe}	Vecteur sollicitation
A	Section transversale
(i)	points soumis à des forces concentrées.
{f ^s }	Forces de surface.
{f ^v }	Forces de volume.
{F}	Forces concentrée.
{u} et {u ^s }	déplacements correspondant aux forces {f ^s } et {f ^v }:
L	longueur d'élément barre ou poutre
I	inertie de l'élément poutre
Q4	élément quadratique linéaire à 4 Nœuds
Q8	élément quadratique isoparamétrique à 8 Nœuds
T3	élément triangulaire linéaire à 3 Nœuds
T6	élément triangulaire isoparamétrique à 6 Nœuds
T4	élément tétraèdre linéaire à 4 Nœuds
C8	élément hexaèdre cubique linéaire à 8 Nœuds
C20	élément hexaèdre cubique isoparamétrique à 20 Nœuds

Abréviations

ARSAP	Autodesk Robot Structural Analysis Professionnal
MATLAB	Matrix Laboratory
GUI	Graphical User Interface
M.E.F	Méthode des éléments finis
P.T.V	Principe des Travaux Virtuels
R.D.M	Résistance des Matériaux
ref []	Référence Bibliographique

Introduction Générale

Contexte de l'étude

La méthode des éléments finis est aujourd'hui la méthode de calcul la plus utilisée dans les problèmes d'ingénierie. Dans cette technique numérique, toutes les complexités d'un problème telles que la forme, les limites et les conditions de chargement restent les mêmes, mais les résultats obtenus sont approximatifs et dépendent de l'éléments utilisé et du raffinement du maillage.

Lors de l'utilisation de cette méthode, les calculs sont robustes en raison du grand nombre d'inconnues conduisant à une grande pile d'équations simultanées à résoudre par l'utilisateur. Par conséquent, l'utilisation de programmes informatiques pour prendre soin de ces équations est un aspect de la méthode.

L'évaluation de ces équations simultanées se fait en utilisant l'une des méthodes matricielles de résolution. Le langage de programmation MATLAB est utile pour illustrer la programmation de la méthode des éléments finis car il permet de coder très rapidement des méthodes numériques et dispose d'une vaste bibliothèque mathématique prédéfinie adaptée à la manipulation des matrices.

On peut classer les problèmes de structures en traitements :

- Linéaires statiques
- Linéaires dynamiques
- Non - linéarité géométrique
- Non - linéarité matérielle

Cadre et objectif du projet de fin d'étude :

L'objectif de ce mémoire est l'élaboration d'un programme en éléments finis

Il existe plusieurs sortes de formulations d'éléments finis en mécanique des structures, formulation contraintes, formulation déplacement ou formulation mixtes....etc. Mais la plus utilisée et qu'on développera dans notre étude est la formulation déplacements vu son efficacité numérique et parce qu'elle permet la résolution aisée de problèmes linéaires qui sont l'objet de notre projet.

On formule le problème en fonction des déplacements aux nœuds et les différentes étapes de calcul sont les suivantes :

- Idéalisation / modélisation : choix du modèle mathématique (champ de déplacement).
- Discrétisation : maillage.
- Choix des hypothèses sur le champ de déplacement dans les éléments (type d'élément) et interpolation du déplacement dans l'élément fini en fonction des inconnues nodales qui conduit à l'établissement des matrices de rigidités élémentaires.
- Formation de la matrice de rigidité de la structure tout entière en vue d'écriture des équations d'équilibre aux nœuds.
- Formation du vecteur force nodale correspondant aux diverses charges appliquées.
- Introduction des conditions aux limites de déplacement permettant la détermination des déplacements nodaux de la structure.
- Résolution des équations d'équilibre pour déterminer le vecteur des déplacements nodaux (inconnus nodaux).
- Substitution des déplacements des nœuds pour construire : le champ de déformation, le champ de contrainte.

Organisation du manuscrit

- Dans le présent travail on se propose une introduction générale suivit du premier chapitre consacré à l'étude bibliographique sur la théorie de l'élasticité linéaire.
- Le deuxième chapitre met en évidence la méthode des éléments finis, appliquée au calcul des structures linéaires et sa formulation.
- Le troisième chapitre exposera une présentation du programme de calcul **Apl_Design_MARS** version1.0, développé sous Matlab
- Le quatrième chapitre est consacré à La validation du programme et calcul numérique sera l'objet du cinquième chapitre.
- Finalement, l'étude sera finalisée par une conclusion générale sur le sujet d'étude et des recommandations.

Chapitre 1

Étude bibliographique

Sommaire

1.	Introduction	19
1.1.	Introduction	19
1.2.	Equation de base de la théorie de l'élasticité linéaire.....	19
1.2.1.	Équations d'équilibre (03 eq) :	19
1.2.2.	Equations de compatibilité (06 eq) :	20
1.2.3.	Equations définissant la loi de comportement (06 eq) :	20
1.3.	Relation contraintes déformations :	21
1.3.1.	Elasticité plane	21
1.3.1.1.	Contrainte plane	22
1.3.1.2.	Déformation plan	22
1.4.	Méthodes de résolutions	23
1.4.1.	Méthode des déplacements	23
1.4.2.	Organigramme de calcul	24
1.4.3.	Minimisation de l'énergie potentielle totale	25

1.1. Introduction

La résolution d'un problème linéaire de structure en utilisant la méthode des éléments finis nécessite la connaissance des équations de bases de la théorie de l'élasticité linéaire

1.2. Equation de base de la théorie de l'élasticité linéaire.

Un solide en équilibre soumis à des forces de volume ainsi qu'à des déplacements ou des tractions imposés sur sa frontière est régit par le système d'équations composés de 15 équations et donc 15 inconnues (trois composantes de déplacement u_i , six composantes de déformation ε_{ij} , et six composantes de contrainte σ_{ij})

Ainsi les trois champs vectoriels à étudier sont Les champs de déplacements, contraintes et déformations.

$$u(x) = \begin{cases} u(x, y, z) \\ v(x, y, z) \\ w(x, y, z) \end{cases} \quad \sigma_{ij} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \quad \varepsilon_{ij} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{bmatrix} \quad (1.01)$$

Champs de déplacement

Champs de contraintes

Champs de déformations

1.2.1. Équations d'équilibre

$$\sigma_{ij,j} + F_i = 0 \quad (1.02)$$

Sous forme développée

$$\frac{\partial \sigma_{11}}{\partial x_1} + \frac{\partial \sigma_{12}}{\partial x_2} + \frac{\partial \sigma_{13}}{\partial x_3} = F_1 \quad (1.03)$$

$$\frac{\partial \sigma_{12}}{\partial x_1} + \frac{\partial \sigma_{22}}{\partial x_2} + \frac{\partial \sigma_{23}}{\partial x_3} = F_2 \quad (1.04)$$

$$\frac{\partial \sigma_{13}}{\partial x_1} + \frac{\partial \sigma_{23}}{\partial x_2} + \frac{\partial \sigma_{33}}{\partial x_3} = F_3 \quad (1.05)$$

1.2.2. Equations de compatibilité

$$\frac{\partial^2 \varepsilon_{11}}{\partial X_2^2} + \frac{\partial^2 \varepsilon_{22}}{\partial X_1^2} - \frac{\partial^2 \varepsilon_{12}}{\partial X_1 \partial X_2} = 0 \quad (1.06)$$

$$\frac{\partial^2 \varepsilon_{22}}{\partial X_3^2} + \frac{\partial^2 \varepsilon_{33}}{\partial X_2^2} - \frac{\partial^2 \varepsilon_{23}}{\partial X_2 \partial X_3} = 0 \quad (1.07)$$

$$\frac{\partial^2 \varepsilon_{33}}{\partial X_1^2} + \frac{\partial^2 \varepsilon_{11}}{\partial X_3^2} - \frac{\partial^2 \varepsilon_{31}}{\partial X_3 \partial X_1} = 0 \quad (1.08)$$

$$\frac{\partial^2 \varepsilon_{11}}{\partial X_2 \partial X_3} + \frac{\partial}{\partial X_1} \left(\frac{\partial \varepsilon_{23}}{\partial X_1} - \frac{\partial \varepsilon_{31}}{\partial X_2} - \frac{\partial \varepsilon_{12}}{\partial X_3} \right) = 0 \quad (1.09)$$

$$\frac{\partial^2 \varepsilon_{22}}{\partial X_3 \partial X_1} + \frac{\partial}{\partial X_2} \left(\frac{\partial \varepsilon_{31}}{\partial X_2} - \frac{\partial \varepsilon_{12}}{\partial X_3} - \frac{\partial \varepsilon_{23}}{\partial X_1} \right) = 0 \quad (1.10)$$

$$\frac{\partial^2 \varepsilon_{33}}{\partial X_1 \partial X_2} + \frac{\partial}{\partial X_3} \left(\frac{\partial \varepsilon_{12}}{\partial X_3} - \frac{\partial \varepsilon_{23}}{\partial X_1} - \frac{\partial \varepsilon_{31}}{\partial X_2} \right) = 0 \quad (1.11)$$

1.2.3. Equations définissant la loi de comportement (06 eq)

$$\sigma_{ij} = \lambda \text{ trace } \varepsilon_{ij} + 2G \varepsilon_{ij} = \frac{\nu E}{(1 + \nu)(1 - 2\nu)} \text{ trace } \varepsilon_{ij} + \frac{E}{1 + \nu} \varepsilon_{ij} \quad (1.12)$$

$$\varepsilon_{ij} = \frac{-\lambda}{2G(3\lambda + 2G)} \text{ trace } \sigma_{ij} \delta_{ij} + \frac{1}{2G} \sigma_{ij} = -\frac{\nu}{E} \text{ trace } \sigma_{ij} \delta_{ij} + \frac{1 + \nu}{E} \sigma_{ij} \quad (1.13)$$

Où λ et μ sont les constantes de Lamé.

G : Est le module de cisaillement.

E : Module de Young

ν : Coefficient de Poisson

$$\delta_{ij} = \begin{cases} 0 & \text{si } i \neq j \\ 1 & \text{si } i = j \end{cases} \quad (1.14)$$

Le symbole de Kronecker est défini par

Les relations entre les paramètres de comportements sont les suivantes

$$\begin{array}{cccc} \mu = \frac{E}{2(1+\nu)} & \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} & E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu} & \nu = \frac{\lambda}{2(\lambda + \mu)} \\ \text{Constantes} & \text{De Lamé} & \text{Module de Young} & \text{Coefficient de Poisson} \\ & & & (1.15) \text{ à } (1.18) \end{array}$$

1.3. Relation contraintes déformations :

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{Bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} \quad (1.19)$$

1.3.1. Elasticité plane

L'élasticité est dite plane lorsque l'étude des contraintes et des déformations peut se ramener à des problèmes à deux dimensions.

$$[\sigma] = \begin{bmatrix} \sigma_x & \sigma_{xy} & 0 \\ \sigma_{yx} & \sigma_y & 0 \\ 0 & 0 & \sigma_z \end{bmatrix} \quad (1.20)$$

$$[\varepsilon] = \begin{bmatrix} \varepsilon_x & \varepsilon_{xy} & 0 \\ \varepsilon_{yx} & \varepsilon_y & 0 \\ 0 & 0 & \varepsilon_z \end{bmatrix} \quad (1.21)$$

1.3.1.1. Contrainte plane

L'hypothèse des contraintes planes est surtout utilisée pour modéliser des corps élastiques planes et minces chargés dans leur plan.

On peut citer comme exemples de structures étudiés en contraintes planes :

- Plaque
- Voiles minces

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E}{(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \quad (1.22)$$

1.3.1.2. Déformation plan

L'hypothèse des déformations planes est surtout utilisée pour les corps élastiques longs dans la direction z , de section de forme quelconque mais constante suivant z , chargés parallèlement au plan des sections de la même manière dans toutes les sections. On citera comme exemples étudiés en déformations planes

- Les talus - Les murs de soutènement
- Les tunnels - Les barrages poids

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \quad (1.23)$$

1.4. Méthodes de résolutions

Il existe trois méthodes de résolution des problèmes en élasticité linéaire

- La méthode de L'AME & NAVIER formulation qui prend pour inconnues les déplacements est une méthode exacte
- La méthode de BELTRAMI ou formulation en contrainte dont les inconnues sont les contraintes.
- La méthode mixte

Dans notre travail on s'intéressera à la méthode des déplacements dite aussi approche cinématique

1.4.1. Méthode des déplacements

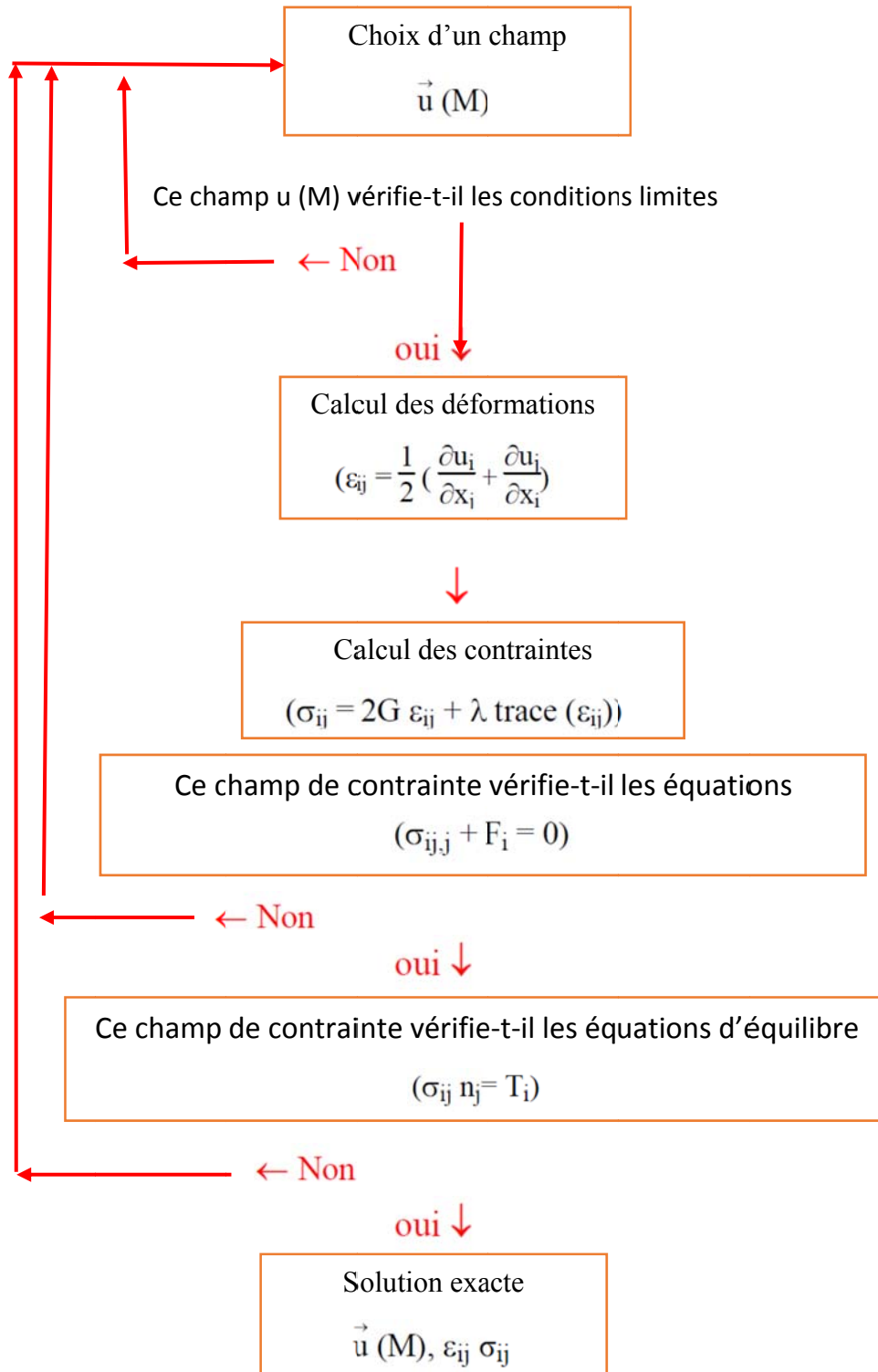
Appelé aussi **Méthode de L'AME & NAVIER** ou formulation en déplacement basée sur le principe de minimisation de l'énergie potentielle totale avec cette méthode ou on choisit comme inconnues les 03 trois composantes du vecteur de déplacements et on exprime en fonction d'elles les 12 autres inconnues du problème

$$\varepsilon = 1/2 (\text{grad } \vec{u} + \text{grad}^T \vec{u}) \quad (1.24)$$

$$\sigma = \lambda (\text{div } \vec{u}) \mathbf{I} + G (\text{grad } \vec{u} + \text{grad}^T \vec{u}) \quad (1.25)$$

$$\overline{\text{div}} \sigma = (\lambda + G) \cdot \text{grad}(\vec{u} \cdot \text{div } \vec{u}) + G \cdot \overline{\Delta} \vec{u} = -\vec{f} \quad (1.26)$$

1.4.2. Organigramme de calcul



1.4.3. Minimisation de l'énergie potentielle totale

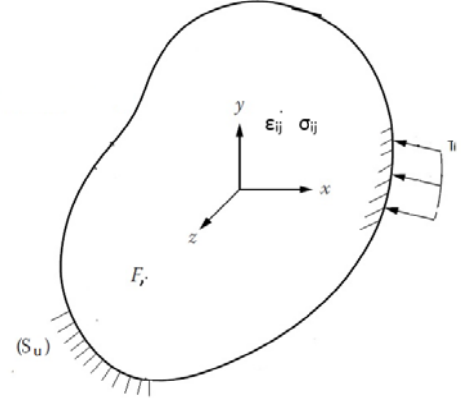
Le travail des efforts extrêmes

$$w_e = \int_V \vec{u} \vec{f} dv + \int_{ST} \vec{u} \vec{T} ds \quad (1.27)$$

$$\varepsilon_{ij} = \frac{1}{2} (\delta u_{ij} + \delta u_{ji}) \quad (1.28)$$

$$Ed = \frac{1}{2} \int_V \varepsilon^T \sigma dv = \frac{1}{2} \int_V \varepsilon^T D \varepsilon dv \quad (1.29)$$

$$\delta Ed = \frac{1}{2} \left[\int_V \delta \varepsilon^T D \varepsilon dv + \int_V \varepsilon^T D \delta \varepsilon dv \right] \quad (1.30)$$



puisque D est symétrique les deux termes sont égaux

$$\delta Ed = \int_V \varepsilon^T D \delta \varepsilon dv = \int_V \delta \varepsilon^T \sigma dv \quad (1.31)$$

La variation du travail des efforts extérieurs dans le champ de déplacement s'écrit

$$\delta w_e = \int_V \delta \vec{u} \vec{f} dv + \int_{ST} \delta \vec{u} \vec{T} ds \quad (1.32)$$

la première variation de la fonction

$$\delta \Pi = \delta Ed - \delta w_e \quad (1.33)$$

$$\delta \Pi = \int_V \delta \varepsilon^T \sigma dv + \int_V \delta \vec{u} \vec{F} dv + \int_{ST} \delta \vec{u} \vec{T} ds = 0 \quad (1.34)$$

on a le Principe du travail virtuel

$$\int_V \sigma_{ij} \varepsilon_{ij} dv = \int_V F_i U_i dv + \int_{ST} T_i U_i ds \quad (1.35)$$

On introduit δ

$$\int_V \sigma_{ij} \varepsilon_{ij} dv = \int_V F_i U_i dv + \int_{ST} T_i U_i ds \quad (1.36)$$

On comparant (1.34) et (1.36) on remarque que $\delta = 0$ et donc la fonction passe par un extremum

$$\delta^2 \Pi > 0 \quad (1.37)$$

$$\delta^2 \Pi = \delta \int_V \delta \varepsilon^T \sigma dv + \delta \int_V \delta \vec{u} \vec{F} dv + \delta \int_{ST} \delta \vec{u} \vec{T} ds = 0 \quad (1.38)$$

$$\delta^2 \Pi = \int_V \delta \delta \varepsilon^T \sigma dv + \int_V \delta \varepsilon^T D \delta \varepsilon dv - \int_V \delta \delta \vec{u} \vec{F} dv - \delta \int_{ST} \delta \delta \vec{u} \vec{T} ds = 0 \quad (1.39)$$

$$\delta^2 \Pi = + \int_V \delta \varepsilon^T D \delta \varepsilon dv > 0 \quad (1.40)$$

D : est une matrice symétrique définie positive
D'où

$$\delta \varepsilon^T \delta \varepsilon > 0 \quad (1.41)$$

$$\delta^2 \Pi > 0 \quad (1.42)$$

Π passe par un minimum parmi tous les champs des déplacements cinématiques admissibles

Π

$$\varepsilon_{ij} = \frac{1}{2} (\delta u_{ij} + \delta u_{ji}) \quad (1.43)$$

$$U_{ij} = \frac{\partial U_j}{\partial_i} \quad (1.44)$$

Le champ de déplacement réel solution minimise l'énergie potentielle totale

Chapitre 2

Méthode des éléments finis

Sommaire

2.	Introduction	29
2.1	Introduction	29
2.2	Discrétisation du milieu	30
2.3	Types d'éléments finis	32
2.4	Fonctions de forme	35
2.5	Application au principe des travaux virtuels :	36
2.6	Choix des éléments programmés	37
2.7	Formulation des éléments finis	38
2.7.1	Interpolation de déplacement :	38
2.7.2	Interpolation de la déformation :	38
2.7.3	Interpolation de la contrainte :	38
2.7.4	Matrice de rigidité élémentaire :	39
2.8	Éléments de référence 1D	40
2.8.1	Élément barre.....	40
2.9	Éléments de référence 2D	41
2.9.1	Élément Poutre	41
2.10	Éléments isoparamétriques	42
2.10.1	Élément triangulaire à 6 nœuds T6	42
2.10.2	Élément quadratique iso-paramétrique à huit nœuds	44
2.11	Éléments de référence 3D.....	47
2.11.1	Élément Tétraèdre a 4 nœuds.....	48
2.11.2	Élément hexaèdre a 8 nœuds.....	50
2.11.3	Élément hexaèdre a 20 nœuds.....	54
2.12	Le vecteur de force.....	57
2.13	Conditions aux limites :	57
2.14	Méthode de factorisation de Cholesky	58
2.15	Intégration numérique	60

2.1 Introduction

La méthode des éléments finis permet de ramener les problèmes des milieux continus régis par des équations différentielles à des problèmes discrets aux nombre fini de paramètres inconnus.

Dans ce chapitre nous allons voir les différentes phases de formulation d'un problème d'éléments finis en statique est en dynamique, l'expression des différents termes qui permettent la formulation des équations de mouvement.

En statique l'équation de mouvement a la forme suivante : $[KG] \times \{X\} = \{Fe\}$ où on a besoin de calculer ($[Ke]$, ($[KG]$, $\{Fe\}$, $\{X\}$) , la matrice de rigidité élémentaire de chaque élément finis, l'assemblage de ces matrices de rigidité, le calcul du vecteur forces extérieures et tenir compte des conditions aux limite, en fin le choix d'une méthode de résolution du système d'équation répondant à la particularité de la matrice de rigidité globale qui est symétrique et définie positive.

Après calcul des déplacements on passe au calcul des contraintes et déformations (σ , ϵ). Qui permettent de continuer l'étude.

Dans ce qui suit nous allons voir quelques notions de base qui permettent de bien faire ce travail tel que :

- **Discrétisation d'un milieu.**
- **Types d'éléments finis.**
- **Fonctions de forme.**
- **Application du Principe des travaux virtuels.**

2.2 Discrétisation du milieu

Comme déjà citée dans l'introduction la MEF est basée sur une idée simple : subdiviser soit discrétiser une forme complexe en un grand nombre de sous-domaines élémentaires de forme géométrique simple (éléments finis) interconnectés en des points appelés nœuds.

Nous considérons le comportement mécanique de chaque élément séparément, puis nous assemblons ces éléments de telle façon que l'équilibre des forces et la compatibilité des déplacements soient satisfaits en chaque nœud.

La MEF utilise des approximations simples des variables inconnues dans chaque élément pour transformer les équations aux dérivées partielles en équations algébriques.

Les nœuds et les éléments n'ont pas forcément de signification physique particulière, mais sont basés sur des considérations de précision de l'approximation.

La première étape de l'analyse par éléments finis implique la division du corps en élément plus petits, appelés éléments finis. Cela équivaut à remplacer le corps par un système ayant un nombre fini de degrés de liberté.

Les formes, le nombre et les configurations des éléments sont choisis de manière à ce que le résultat ressemble le plus possible à celui du corps d'origine. Le choix du type de l'élément est dicté par la géométrie du corps et le nombre de coordonnées indépendantes nécessaire pour décrire le système. Dans l'analyse de déformation simple, la géométrie, les propriétés du matériau et la variable de champ du système peut être décrite en termes de deux ou de trois coordonnées spatiales, c'est-à-dire x, y et z.

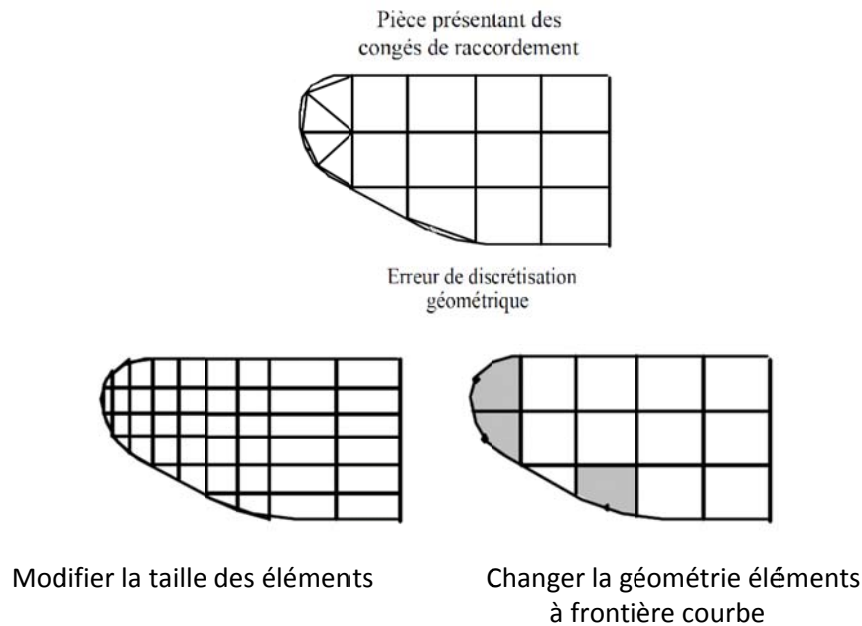
Les éléments sont considérés comme interconnectés en des points spécifiques appelés nœuds.

Les nœuds sont les points finis sélectionnés auxquels les inconnues de base doivent être déterminées dans l'analyse des éléments finis.

. La discrétisation du milieu continu en sous domaines consiste à décomposer le domaine continu en un nombre fini de sous domaines dit éléments finis

$$D = \sum_{e=1}^{ne} D_e \quad \text{telle que} \quad \lim_{\substack{\text{taille des} \\ e \rightarrow 0}} \left(\bigcup_e D_e \right) = D \quad (2.01)$$

Sans qu'il y est aucun recouvrement ou vide entre deux éléments ayant une frontière commune. et lorsque la frontière du domaine est complexe il y a lieu de réduire la taille ou de modifier la forme des éléments concernés afin de minimiser les erreurs de discrétisation géométrique éventuelles et inévitables.



La méthode des éléments finis est basée sur la construction systématique d'une approximation du champ des variables par sous domaine. Cette approximation est construite sur les valeurs approchées du champ aux nœuds de l'élément. On parle alors d'approximation nodale.

L'approximation par éléments finis est une **approximation nodale** par sous domaines ne faisant intervenir que les **variables nodales** du domaine élémentaire.

$$D_e. \quad \forall M \in D_e \quad \{u^h(M)\} = [N(M)] \{U_n\} \quad (2.02)$$

avec

- $\{u^h\}$ Valeur de la fonction approchée en tout point M de l'élément
- $[N]$ Matrice des fonctions d'interpolation de l'élément
- $\{U_n\}$ Variables nodales relatives aux nœuds d'interpolation de l'élément

2.3 Types d'éléments finis

Il existe plusieurs types d'éléments finis, employées dans différents contextes. La notion de type d'élément est liée à leur principe de fonctionnement ou chaque élément est muni d'une géométrie de référence figée, dans laquelle sa forme exacte, la position de ses nœuds, ses fonctions de base et ses degrés de liberté sont spécifiés. L'ensemble de ces données s'appelle éléments de référence.

De manière générale un élément fini est défini par quatre données :

1. Un domaine géométrique dont la dimension dépend de la théorie considérée, comme par exemple un polyèdre, un polygone ou un segment ;
2. Un ensemble de points situés dans ce domaine ou sur son bord, nommés nœuds ;
3. Un ensemble de fonctions, chacune associée à un nœud, définies sur le domaine et à valeurs réelles, nommées fonctions de base ou de forme ;
4. Une hypothèse cinématique consistant à exprimer une grandeur physique comme par exemple un champ de déplacement (contraintes ou déformations planes) à l'intérieur du domaine sous forme d'une combinaison linéaire des fonctions de base, dont les coefficients peuvent être quelconques ; chaque terme de cette combinaison linéaire est appelé degré de liberté.

Afin de réduire le nombre d'opérations à effectuer, les logiciels ramènent tous les calculs sur les éléments de référence. Pour cela, chaque élément « réel » du maillage est supposé issu d'une transformation géométrique appliquée à un élément de référence, c'est-à-dire au domaine géométrique, aux nœuds, au support des fonctions de base... On dit que deux éléments « réels » sont de même type s'ils sont issus du même élément de référence par des transformations éventuellement différentes : un type d'élément désigne donc l'ensemble des éléments pouvant être obtenus par transformation d'un élément de référence donné.

Cette définition suppose que la nature de la transformation soit elle-même une caractéristique intrinsèque de l'élément de référence. La plupart des logiciels emploient des éléments, dits iso-paramétriques, qui vérifient cette condition car les transformations que l'on peut appliquer à leur géométrie de référence sont, par construction, de même nature que les champs de déplacement qu'ils peuvent

représenter. Par exemple, un élément capable de représenter un champ de déplacement affine sera issu de sa géométrie de référence par une transformation elle aussi affine.

En 2D :

- triangles
 - ✓ triangles de degré 1, (triangles à 3 nœuds, fonctions linéaires)
 - ✓ triangles de degré 2 (triangles à 6 nœuds, polynômes de degré 2)
 - ✓ triangles de degré 3 (triangles à 9 nœuds, polynômes de degré 3)
- quadrilatères
 - ✓ quadrilatères de degré 1 (carrés à quatre nœuds, fonctions linéaires)
 - ✓ quadrilatères de degré 2 (carrés à 8 ou 9 nœuds, polynômes de degré 2)
 - ✓ quadrilatères de degré 3 (carrés à 12 ou 13 nœuds, polynômes de degré 3)

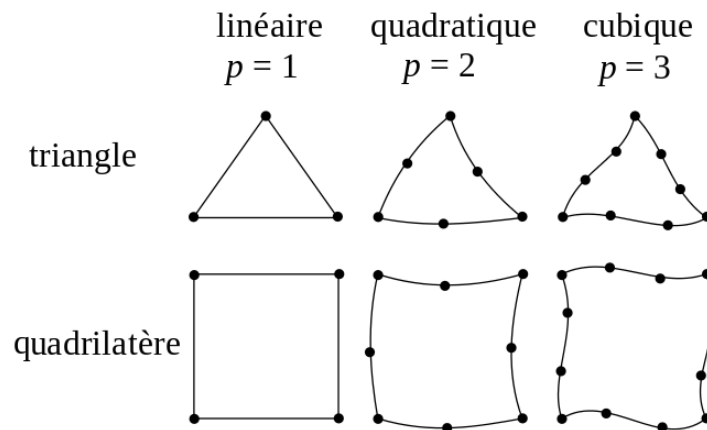


Figure 1 Principaux types d'éléments utilisé en 2D

Il existe deux types de nœuds, les nœuds externes et les nœuds internes. Les nœuds externes sont ceux qui se produisent sur les bords ou la surface des éléments et ils peuvent être communs à deux ou plusieurs éléments. Ces nœuds peuvent être classés en outre comme nœuds primaires et secondaires. Les nœuds primaires se trouvent aux coins des éléments. Les nœuds secondaires se trouvent le long du côté d'un élément mais pas aux coins. Les nœuds internes sont ceux qui se produisent à l'intérieur d'un élément et ils sont uniques à chaque élément.

En 3D :

- tétraèdres
 - ✓ tétraèdres de degré 1, (quatre nœuds, fonctions linéaires)
 - ✓ tétraèdres de degré 2, (dix nœuds, polynômes de degré 2)

- hexaèdres
 - ✓ hexaèdres de degré 1, (huit nœuds, fonctions linéaires)
 - ✓ hexaèdres de degré 2, (vingt nœuds, polynômes de degré 2)
 - ✓ hexaèdres triquadratique, (vingt-sept nœuds, polynômes de degré 2)

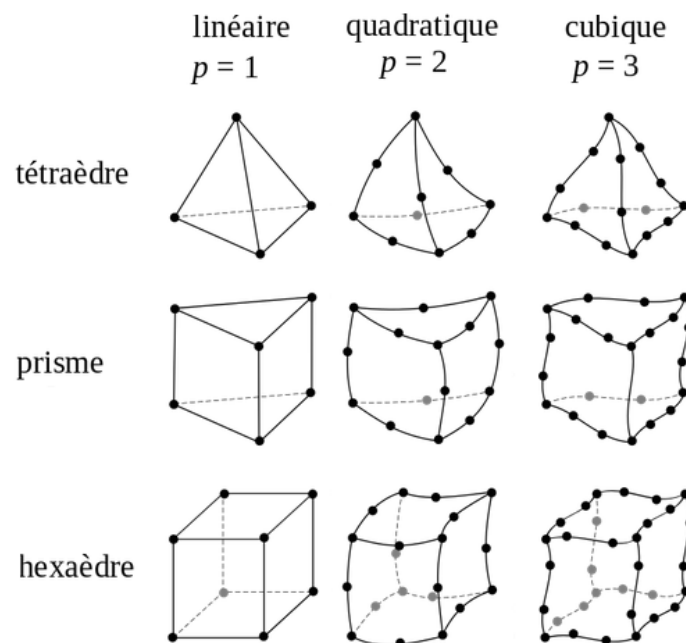


Figure 02 : Principaux types d'éléments utilisé en 3D

2.4 Fonctions de forme

Dans l'analyse par éléments finis, le but est de trouver les variables de champ aux points nodaux par une analyse rigoureuse, en supposant qu'en tout point à l'intérieur de la variable de base de l'élément est une fonction des valeurs aux points nodaux de l'élément. Cette fonction qui relie la variable de champ en tout point de l'élément aux variables de champ des points nodaux est appelée fonction de forme ou fonction d'interpolation. Ref[18]

En prenant le déplacement comme variable de champ, cette relation peut être exprimée comme suit

$$u = \sum_{i=1}^n N_i u_i \quad (2.03)$$

$$v = \sum_{i=1}^n N_i v_i \quad (2.04)$$

Où

u = déplacement horizontal

v = déplacement vertical

u_i = déplacement horizontal au nœud i

v_i = déplacement vertical au nœud i

N_i = expression de la fonction de forme au nœud i

Les fonctions de forme sont toujours exprimées en termes de système de coordonnées naturel. Un système de coordonnées naturelles est un système de coordonnées qui permet la spécification d'un point dans l'élément par un ensemble de nombres sans dimension, dont la magnitude ne dépasse jamais l'unité. Il est obtenu en attribuant des poids aux coordonnées nodales lors de la définition de la coordonnée de tout point à l'intérieur de l'élément. Par conséquent, un tel système a la propriété que la i ème coordonnée a une valeur unitaire au nœud i de l'élément et une valeur nulle à tous les autres nœuds.

Ainsi Les fonctions d'interpolation satisfont la propriété suivante

$$\forall M_i \quad N_{j(M_i)} = \begin{cases} 0 & \text{si } i \neq j \\ 1 & \text{si } i = j \end{cases} \quad (2.05)$$

2.5 Application du principe des travaux virtuels :

Pour un corps solide en équilibre, et pour tous champs de déplacements virtuels cinématiquement admissible et satisfaisant les conditions aux limites de déplacement le travail virtuel interne totale de déformation doit être égale au travail virtuel des forces extérieures.

$$\int_V \delta \{\boldsymbol{\varepsilon}\}^T \{\boldsymbol{\sigma}\} dV = \int_V \delta \{u\}^T \{f^v\} dV + \int_{S_r} \delta \{u^s\}^T \{f^s\} dS + \sum_i \delta \{u^i\}^T \{F^i\} \quad (2.06)$$

i : points soumis à des forces concentrées.

$\{f^s\}$: Forces de surface.

$\{f^v\}$: Forces de volume.

$\{F\}$: Forces concentrée.

$\{u\}$ et $\{u^s\}$: Sont les déplacements correspondant aux forces $\{f^s\}$ et $\{f^v\}$:

L'énergie potentielle totale Π est stationnaire de plus pour un équilibre stable elle est minimum :

$$\Pi = U + V \quad (2.07)$$

U : Energie de déformation interne

V : Potentiel des forces appliquées (opposé du travail des forces extérieures).

$$\Pi = \frac{1}{2} \int_V \{\boldsymbol{\varepsilon}\}^T \{\boldsymbol{\sigma}\} dV - \int_V \{u\}^T \{f^v\} dV - \int_{S_r} \{u^s\}^T \{f^s\} dS - \sum_i \{u_i\}^T \{F_i\} \quad (2.08)$$

2.6 Choix des éléments programmés

Dans notre travail le choix d'élément finis concernée par notre programme est comme suite :

- Problèmes à une dimension :
Élément barre linéaire, élément poutre.

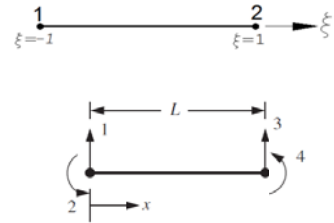


Figure 3 : type d'éléments étudiés 1D

- Problèmes à deux dimensions :
Élément triangle T6, élément quadratique Q8

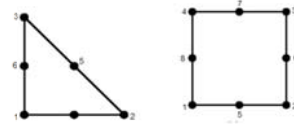


Figure 4 : Type d'éléments étudiés 2D

- Problèmes trois dimensions :
Élément cubique C8, C20
Et T4 le tétraèdre à quatre nœuds

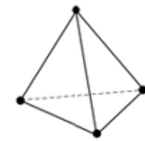
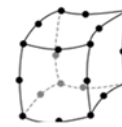
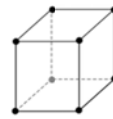


Figure 5 : Type d'éléments étudiés 3D

2.7 Formulation des éléments finis

Tenant compte des notions déjà citées au-dessus nous allons aborder la formulation des différentes grandeurs suivantes : déplacements, déformations, contraintes, et matrice de rigidité.

En suite pour chaque élément choisi dans notre programme un calcul détaillé sera élaboré pour chaque une de ces grandeurs.

2.7.1 Interpolation de déplacement :

Elle permet d'exprimer le déplacement en tout point de l'élément en fonction du déplacement des nœuds.

$$\{u\} = [N_e]\{u_e\} \quad (2.09)$$

$[N_e]$: matrice ligne de fonction d'interpolation ou de forme

2.7.2 Interpolation de la déformation :

Les déformations de l'élément ' e ' sont obtenus par différentiation des déplacements :

$$\{\varepsilon_e\} = [L][N_e]\{u_e\} \quad (2.10)$$

avec $\{\varepsilon_e\} = [B_e]\{u_e\} \quad (2.11)$

$$[B_e] = [L][N_e] \quad (2.12)$$

$[L]$: Matrice Laplacienne.

2.7.3 Interpolation de la contrainte :

Les contraintes sont liées aux déformations par la relation :

$$\{\sigma_e\} = [D]\{\varepsilon_e\} + \{\sigma_0^e\} \quad (2.13)$$

où $\{\sigma_0^e\}$: les contraintes initiales. Et $[D]$: matrice d'élasticité.

2.7.4 Matrice de rigidité élémentaire :

D'après le théorème des travaux virtuels on a :

$$W_{int} = W_{ext} \quad (2.14)$$

$$\int_V \{\varepsilon_e\}^T \{\sigma_e\} dV = \{\tilde{u}_e\}^T \{F_e\} \quad (2.15)$$

$$\int_V ([B_e] \{u_e\})^T ([D][B_e] \{u_e\}) dV = \{\tilde{u}_e\}^T \{F_e\} \quad (2.16)$$

$$\{\tilde{u}_e\}^T \left[\int_V [B]^T [D][B] dV \right] \{u_e\} = \{\tilde{u}_e\}^T \{F_e\} \quad (2.17)$$

$$\left[\int_V [B_e]^T [D][B_e] dV \right] \{u_e\} = \{F_e\} \quad (2.18)$$

$\{\tilde{u}_e\}$: déplacement virtuel.

la matrice de rigidité élémentaire sous la forme

$$[K_e] = \int_V [B_e]^T [D][B_e] dV \quad (2.19)$$

L'expression des forces énergiquement équivalentes aux nœuds s'écrit

$$\{F^e\} = \int_{V_e} [N]^T \cdot \{b\} dv - \int_{S_e} [N]^T \cdot \{q\} ds - \{F\} \quad (2.20)$$

forces de volume (poids propre)

$$\int_{V_e} [N]^T \cdot \{b\} dv \quad (2.21)$$

forces surfaciques

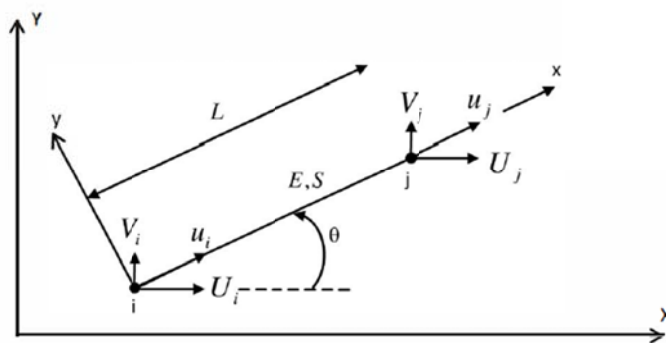
$$\int_{S_e} [N]^T \cdot \{q\} ds \quad (2.22)$$

$\{F\}$: forces nodales

2.8 Éléments de référence 1D

2.8.1 Élément barre

L'élément de barre linéaire est un élément fini unidimensionnel où les coordonnées locales et globales coïncident. Il se caractérise par des fonctions de forme linéaire et est identique à l'élément ressort sauf que la rigidité de la barre n'est pas donnée directement. L'élément linéaire soumis qu'à des forces axiales donc ne peuvent pas subir de flexion ou de cisaillement ainsi ne peuvent supporter que la compression ou la tension [Ref. \[2\]](#)



E : module d'élasticité
S : section transversale
L : longueur de la barre

Figure 6 : élément Barre

Matrice de rigidité est donnée comme suit

Dans le repère locale

$$[k_e] = \frac{ES}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (2.23)$$

Dans le repère global

$$[K_e] = \frac{ES}{L} \begin{bmatrix} \cos^2 \theta & \cos \theta \sin \theta & -\cos^2 \theta & -\cos \theta \sin \theta \\ \cos \theta \sin \theta & \sin^2 \theta & -\cos \theta \sin \theta & -\sin^2 \theta \\ -\cos^2 \theta & -\cos \theta \sin \theta & \cos^2 \theta & \cos \theta \sin \theta \\ -\cos \theta \sin \theta & -\sin^2 \theta & \cos \theta \sin \theta & \sin^2 \theta \end{bmatrix} \quad (2.24)$$

2.9 Éléments de référence 2D

2.9.1 Élément Poutre

L'élément poutre est un élément fini bidimensionnel où les coordonnées locales et globales coïncident. Il est caractérisé par des fonctions de forme linéaire.

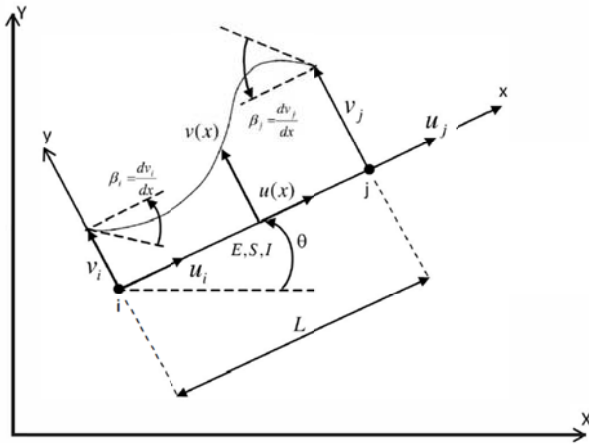


Figure 7 : Élément Poutre

L'élément de poutre a un module d'élasticité E, un moment d'inertie I et une longueur L. Chaque élément de poutre a deux nœuds et est supposé être horizontal comme illustré. [Ref. \[2\]](#)

Matrice de rigidité repère locale

$$[k_e] = \begin{bmatrix} \frac{ES}{L} & 0 & 0 & -\frac{ES}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{ES}{L} & 0 & 0 & \frac{ES}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \quad \begin{array}{l} E : \text{module d'élasticité} \\ S : \text{section transversale} \\ L : \text{longueur de la poutre} \\ I : \text{inertie de la poutre} \end{array} \quad (2.25)$$

Matrice de rigidité repère globale ; [Ref. \[2\]](#)

$$\begin{bmatrix} \frac{ES}{L} \cos^2 \theta + \frac{12EI}{L^3} \sin^2 \theta & \left(\frac{ES}{L} - \frac{12EI}{L^3} \right) \cos \theta \sin \theta & -\frac{6EI}{L^2} \sin \theta & -\frac{ES}{L} \cos^2 \theta - \frac{12EI}{L^3} \sin^2 \theta & \left(-\frac{ES}{L} + \frac{12EI}{L^3} \right) \cos \theta \sin \theta & -\frac{6EI}{L^2} \sin \theta \\ \left(\frac{ES}{L} - \frac{12EI}{L^3} \right) \cos \theta \sin \theta & \frac{ES}{L} \sin^2 \theta + \frac{12EI}{L^3} \cos^2 \theta & \frac{6EI}{L^2} \cos \theta & \left(-\frac{ES}{L} + \frac{12EI}{L^3} \right) \cos \theta \sin \theta & -\frac{ES}{L} \sin^2 \theta - \frac{12EI}{L^3} \cos^2 \theta & \frac{6EI}{L^2} \cos \theta \\ -\frac{6EI}{L^2} \sin \theta & \frac{6EI}{L^2} \cos \theta & \frac{4EI}{L} & \frac{6EI}{L^2} \sin \theta & -\frac{6EI}{L^2} \cos \theta & \frac{2EI}{L} \\ -\frac{ES}{L} \cos^2 \theta - \frac{12EI}{L^3} \sin^2 \theta & \left(-\frac{ES}{L} + \frac{12EI}{L^3} \right) \cos \theta \sin \theta & \frac{6EI}{L^2} \sin \theta & \frac{ES}{L} \cos^2 \theta + \frac{12EI}{L^3} \sin^2 \theta & \left(\frac{ES}{L} - \frac{12EI}{L^3} \right) \cos \theta \sin \theta & \frac{6EI}{L^2} \sin \theta \\ \left(-\frac{ES}{L} + \frac{12EI}{L^3} \right) \cos \theta \sin \theta & -\frac{ES}{L} \sin^2 \theta - \frac{12EI}{L^3} \cos^2 \theta & -\frac{6EI}{L^2} \cos \theta & \left(\frac{ES}{L} - \frac{12EI}{L^3} \right) \cos \theta \sin \theta & \frac{ES}{L} \sin^2 \theta + \frac{12EI}{L^3} \cos^2 \theta & -\frac{6EI}{L^2} \cos \theta \\ -\frac{6EI}{L^2} \sin \theta & \frac{6EI}{L^2} \cos \theta & \frac{2EI}{L} & \frac{6EI}{L^2} \sin \theta & -\frac{6EI}{L^2} \cos \theta & \frac{4EI}{L} \end{bmatrix} \quad (2.26)$$

2.10 Éléments iso-paramétriques

2.10.1 Élément triangulaire à 6 nœuds T6

L'élément triangulaire quadratique est un élément fini bidimensionnel caractérisé par des fonctions de forme quadratique. Chaque triangle quadratique a six nœuds avec deux degrés de liberté dans le plan à chaque nœud, Cet élément peut être utilisé pour des problèmes de contrainte plane ou de déformation plane en élasticité. Avec un module d'élasticité E , un coefficient de Poisson ν et une épaisseur t . Les coordonnées globales des six nœuds sont désignées par (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , (x_4, y_4) , (x_5, y_5) et (x_6, y_6) . L'ordre des nœuds pour chaque élément est important - ils doivent être listés dans le sens antihoraire en commençant par les nœuds d'angle puis les nœuds du milieu. L'aire de chaque triangle doit être positive. [Ref \[3\]](#)

La matrice de rigidité de l'élément pour l'élément triangulaire quadratique s'écrit en termes d'une double intégrale comme suit:

$$[k] = t \iint_A [B]^T [D] [B] dx dy \quad (2.27)$$

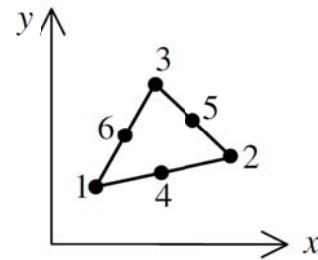


Figure 8 : Nœuds élément triangulaire T6

La matrice D est donnée comme suit

En contrainte plane

$$[D] = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1 - \nu}{2} \end{bmatrix} \quad (2.28)$$

En déformation plane

$$[D] = \frac{E}{(1 + \nu)(1 - 2\nu)} \begin{bmatrix} 1 - \nu & \nu & 0 \\ \nu & 1 - \nu & 0 \\ 0 & 0 & \frac{1 - 2\nu}{2} \end{bmatrix} \quad (2.29)$$

La matrice [B] est donnée comme suit

$$[B] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0 & \frac{\partial N_5}{\partial x} & 0 & \frac{\partial N_6}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_4}{\partial y} & 0 & \frac{\partial N_5}{\partial y} & 0 & \frac{\partial N_6}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} & \frac{\partial N_5}{\partial y} & \frac{\partial N_5}{\partial x} & \frac{\partial N_6}{\partial y} & \frac{\partial N_6}{\partial x} \end{bmatrix} \quad (2.30)$$

Les fonctions de formes

$$\begin{aligned}
 N_1 &= \frac{(x_{23}(y - y_3) - y_{23}(x - x_3))(x_{46}(y - y_6) - y_{46}(x - x_6))}{(x_{23}y_{13} - y_{23}x_{13})(x_{46}y_{16} - y_{46}x_{16})} \\
 N_2 &= \frac{(x_{31}(y - y_1) - y_{31}(x - x_1))(x_{54}(y - y_4) - y_{54}(x - x_4))}{(x_{31}y_{21} - y_{31}x_{21})(x_{54}y_{24} - y_{54}x_{24})} \\
 N_3 &= \frac{(x_{21}(y - y_1) - y_{21}(x - x_1))(x_{56}(y - y_6) - y_{56}(x - x_6))}{(x_{21}y_{31} - y_{21}x_{31})(x_{56}y_{36} - y_{56}x_{36})} \\
 N_4 &= \frac{(x_{31}(y - y_1) - y_{31}(x - x_1))(x_{23}(y - y_3) - y_{23}(x - x_3))}{(x_{31}y_{41} - y_{31}x_{41})(x_{23}y_{43} - y_{23}x_{43})} \\
 N_5 &= \frac{(x_{31}(y - y_1) - y_{31}(x - x_1))(x_{21}(y - y_1) - y_{21}(x - x_1))}{(x_{31}y_{51} - y_{31}x_{51})(x_{21}y_{51} - y_{21}x_{51})} \\
 N_6 &= \frac{(x_{21}(y - y_1) - y_{21}(x - x_1))(x_{23}(y - y_3) - y_{23}(x - x_3))}{(x_{21}y_{61} - y_{21}x_{61})(x_{23}y_{63} - y_{23}x_{63})}
 \end{aligned} \tag{2.31}$$

Avec $x_{ij} = x_i - x_j$ $y_{ij} = y_i - y_j$

A partir de la relation entre les déformations à n'importe quel point de l'élément avec le déplacement nodal

$$\{\varepsilon\} = [B]\{u\} \tag{2.32}$$

$\{\varepsilon\}$ = déformation en tout point de l'élément.

$\{u\}$ = vecteur de déplacement des valeurs nodales pour l'élément

$[B]$ = matrice de déplacement de déformation

le vecteur de contrainte est obtenu pour chaque élément comme suit:

$$\{\sigma\} = [D][B]\{u\} \tag{2.33}$$

où σ est le vecteur de contrainte dans l'élément (de taille 3×1) et u est le vecteur de déplacement d'élément 12×1 . Le vecteur σ est écrit pour chaque élément comme

$$\{\sigma\} = [\sigma_x \sigma_y \tau_{xy}]^T \tag{2.34}$$

2.10.2 Élément quadratique iso-paramétrique à huit nœuds

L'élément quadrilatère quadratique est un élément iso-paramétrique bidimensionnel est caractérisé par des fonctions de forme quadratique dans chacune des directions x et y . Cet élément peut être utilisé pour des problèmes de contrainte plane ou de déformation plane en élasticité. Chaque élément a huit nœuds avec deux degrés de liberté dans le plan pour chaque nœud. L'ordre des nœuds pour chaque élément est important. Ils doivent être listés dans le sens antihoraire en partant des nœuds d'angle suivis des nœuds du milieu. Les huit fonctions de forme de cet élément sont répertoriées explicitement comme suit en termes de coordonnées naturelles ξ et η . [Ref \[3\]](#)

Les fonctions de formes se présentent comme suit

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

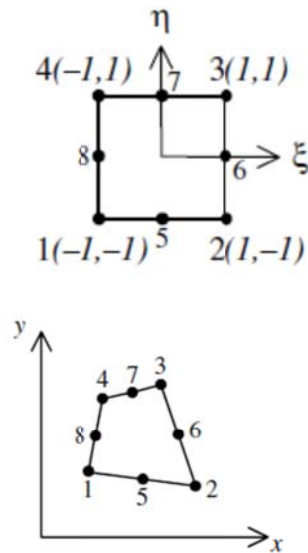


Figure 9 : Coordonnées élément quadratique à huit nœuds

Les fonctions de forme sont exprimées par les coordonnées locales

$$[N] = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 & N_5 & 0 & N_6 & 0 & N_7 & 0 & N_8 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 & N_5 & 0 & N_6 & 0 & N_7 & 0 & N_8 \end{bmatrix} \quad (2.35)$$

Une relation entre les déformations à n'importe quel point de l'élément avec déplacement nodal peut être formée en utilisant des matrices de déplacement de déformation.

$$\{\varepsilon\} = [B]\{u\} \quad (2.36)$$

Ou $[B]$ est la matrice de déplacement de déformation

$$[B] = [L] [N] \quad (2.37)$$

$$[L] = \frac{1}{|J|} \begin{bmatrix} \frac{\partial y}{\partial \eta} \frac{\partial(\cdot)}{\partial \xi} - \frac{\partial y}{\partial \xi} \frac{\partial(\cdot)}{\partial \eta} & & & 0 \\ & 0 & \frac{\partial x}{\partial \xi} \frac{\partial(\cdot)}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial(\cdot)}{\partial \xi} & \\ \frac{\partial x}{\partial \xi} \frac{\partial(\cdot)}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial(\cdot)}{\partial \xi} & \frac{\partial y}{\partial \eta} \frac{\partial(\cdot)}{\partial \xi} - \frac{\partial y}{\partial \xi} \frac{\partial(\cdot)}{\partial \eta} & & \end{bmatrix} \quad (2.38)$$

et la matrice [B] sous sa forme générale

$$[B] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & | & \frac{\partial N_2}{\partial x} & 0 & | & \dots & \dots & | & \frac{\partial N_8}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & | & 0 & \frac{\partial N_2}{\partial y} & | & \dots & \dots & | & 0 & \frac{\partial N_8}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & | & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & | & \dots & \dots & | & \frac{\partial N_8}{\partial y} & \frac{\partial N_8}{\partial x} \end{bmatrix} \quad (2.39)$$

En introduisant la matrice jacobienne qui relie les dérivées de la fonction dans le système de coordonnées naturel local aux dérivées dans le système de coordonnées global

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^8 \frac{\partial N_i}{\partial \xi} x_i & \sum_{i=1}^8 \frac{\partial N_i}{\partial \xi} y_i \\ \sum_{i=1}^8 \frac{\partial N_i}{\partial \eta} x_i & \sum_{i=1}^8 \frac{\partial N_i}{\partial \eta} y_i \end{bmatrix} \quad (2.40)$$

Ou encore

$$[J] = \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \dots & \frac{\partial N_8}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \dots & \frac{\partial N_8}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_8 & y_8 \end{bmatrix} \quad (2.41)$$

Les dérivées par rapport aux coordonnées globales sont calculées avec l'utilisation de l'inverse de la matrice jacobienne

Par conséquent

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix} \quad (2.42)$$

Le déterminant de la matrice jacobienne est utilisé pour la transformation des intégrales du système global de coordonnées au système de coordonnées local:

$$dV = dx dy = |J| d\xi d\eta \quad (2.43)$$

La matrice de rigidité de l'élément pour l'élément quadrilatère quadratique s'écrit en termes d'une double intégrale comme suit:

$$[k] = t \int_{-1}^1 \int_{-1}^1 [B]^T [D] [B] |J| d\xi d\eta \quad (2.44)$$

Il est clair que l'élément quadrilatère quadratique a seize degrés de liberté - deux à chaque nœud. Par conséquent pour une structure à n nœuds, la matrice de rigidité globale K sera de taille $2n \times 2n$ (puisque nous avons deux degrés de liberté à chaque nœud).

$$\{\sigma\} = [D][B] \{u\} \quad (2.45)$$

où σ est le vecteur de contrainte dans l'élément (de taille 3×1) et u est le vecteur de déplacement d'élément 16×1 . Le vecteur σ est écrit pour chaque élément comme

$$\{\sigma\} = [\sigma_x \ \sigma_y \ \tau_{xy}]^T. \quad (2.46)$$

2.11 Éléments de référence 3D

Pour les éléments tridimensionnels hexaédriques de type brique linéaires à 8 nœuds et quadratiques à 20 nœuds, la géométrie et le champ de déplacement sont interpolés avec les fonctions de forme.

$$\begin{aligned} \{u\} &= [N]\{q\} \\ \{u\} &= \{u \ v \ w\} \\ \{q\} &= \{u_1 \ v_1 \ w_1 \ u_2 \ v_2 \ w_2 \ \dots\} \end{aligned} \quad (2.47)$$

Avec

$$[N] = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & \dots \\ 0 & N_1 & 0 & 0 & N_2 & 0 & \dots \\ 0 & 0 & N_1 & 0 & 0 & N_2 & \dots \end{bmatrix} \quad (2.48)$$

Le vecteur de déformation contient six composants différents du tenseur de

$$d^{\text{déformation}} \{\varepsilon\} = \{\varepsilon_x \ \varepsilon_y \ \varepsilon_z \ \gamma_{xy} \ \gamma_{yz} \ \gamma_{zx}\} \quad (2.49)$$

La matrice déformation-déplacement pour les éléments tridimensionnels

$$[B] = [D][N] = [B_1 \ B_2 \ B_3 \ \dots] \quad (2.50)$$

De manière général La matrice Bi est donnée comme suit selon le nombre de nœuds de l'élément élément cube à 8 nœuds ou à 14 nœuds ou a 20 nœuds ou tétraèdre a 4 nœuds

$$[B_i] = \begin{bmatrix} \partial N_i / \partial x & 0 & 0 \\ 0 & \partial N_i / \partial y & 0 \\ 0 & 0 & \partial N_i / \partial z \\ \partial N_i / \partial y & \partial N_i / \partial x & 0 \\ 0 & \partial N_i / \partial z & \partial N_i / \partial y \\ \partial N_i / \partial z & 0 & \partial N_i / \partial x \end{bmatrix} \quad (2.51)$$

Par conséquent

$$\left\{ \begin{array}{l} \partial N_i / \partial x \\ \partial N_i / \partial y \\ \partial N_i / \partial z \end{array} \right\} = [J]^{-1} \left\{ \begin{array}{l} \partial N_i / \partial \xi \\ \partial N_i / \partial \eta \\ \partial N_i / \partial \zeta \end{array} \right\} \quad (2.52)$$

Ainsi la matrice de déformation peut être calculée pour tout type d'élément en utilisant ces expressions

2.11.1 Élément Tétraèdre a 4 nœuds

L'élément tétraédrique linéaire (solide) est un élément fini tridimensionnel caractérisé par des fonctions de forme linéaires. Il est également appelé tétraèdre à déformation constante. Chaque tétraèdre linéaire a quatre nœuds, Les coordonnées globales des quatre nœuds sont désignées par (x_1, y_1, z_1) , (x_2, y_2, z_2) , (x_3, y_3, z_3) et (x_4, y_4, z_4) . La numérotation des nœuds pour chaque élément est très importante. [Ref \[3\]](#)

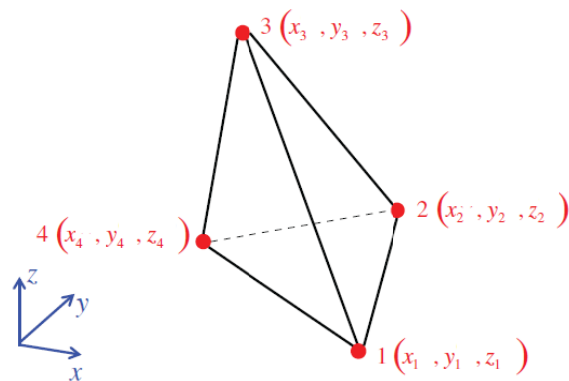


Figure 10 : Coordonnées élément tétraèdre à 4 nœuds

La matrice de rigidité de l'élément est donnée par

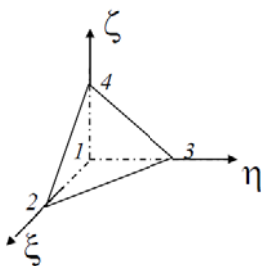
$$[k] = V[B]^T[D][B] \tag{2.53}$$

Avec

$$6V = \begin{vmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix} \tag{2.54}$$

L'élément tétraédrique linéaire a douze degrés de liberté - trois dans chaque nœud. Par conséquent pour une structure à n nœuds, la matrice de rigidité globale K sera de taille $3n \times 3n$

Avec comme fonction de formes



$$\begin{aligned} N_1 &= 1-\xi-\eta-\zeta \\ N_2 &= \xi \\ N_3 &= \eta \\ N_4 &= \zeta \end{aligned} \tag{2.55}$$

La matrice B est donnée

$$[B] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & 0 & \frac{\partial N_2}{\partial x} & 0 & 0 & \frac{\partial N_3}{\partial x} & 0 & 0 & \frac{\partial N_4}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & 0 & \frac{\partial N_2}{\partial y} & 0 & 0 & \frac{\partial N_3}{\partial y} & 0 & 0 & \frac{\partial N_4}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial z} & 0 & 0 & \frac{\partial N_2}{\partial z} & 0 & 0 & \frac{\partial N_3}{\partial z} & 0 & 0 & \frac{\partial N_4}{\partial z} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial z} & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial z} & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial z} & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_4}{\partial z} & \frac{\partial N_4}{\partial y} \\ \frac{\partial N_1}{\partial z} & 0 & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial z} & 0 & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial z} & 0 & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial z} & 0 & \frac{\partial N_4}{\partial x} \end{bmatrix} \quad (2.56)$$

La Matrice [D] s'écrit

$$[D] = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \quad (2.57)$$

Le vecteur de contrainte est obtenu pour chaque élément comme suit:

$$\{\sigma\} = [D][B] \{u\} \quad (2.58)$$

où σ est le vecteur de contrainte dans l'élément de taille 6×1 et u est le vecteur de déplacement de l'élément de taille 12×1 .

Ainsi Le vecteur σ s'écrit pour chaque élément comme suit

$$\{\sigma\} = [\sigma_x \ \sigma_y \ \sigma_z \ \tau_{xy} \ \tau_{yz} \ \tau_{zx}]^T. \quad (2.59)$$

2.11.2 Élément hexaèdre à 8 nœuds

L'élément brique linéaire (solide) est un élément fini tridimensionnel caractérisé par des fonctions de forme linéaire dans chacune des directions x, y et z. Il est également appelé hexaèdre tri linéaire.

L'élément brique linéaire a huit nœuds avec trois degrés de liberté à chaque nœud, Les coordonnées globales des huit nœuds sont notées (x_1, y_1, z_1) , (x_2, y_2, z_2) , (x_3, y_3, z_3) , (x_4, y_4, z_4) , (x_5, y_5, z_5) , (x_6, y_6, z_6) , (x_7, y_7, z_7) et (x_8, y_8, z_8) . L'ordre des nœuds pour chaque élément est important. [Ref \[3\]](#)

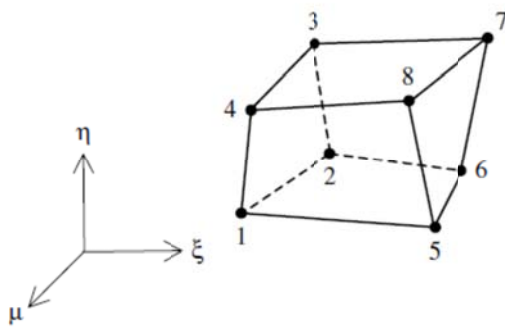


Figure 11 Coordonnées élément cubique à 8 nœuds

Les huit fonctions de forme pour cet élément sont listées explicitement comme suit en termes de coordonnées naturelles ξ , η et μ

$$\begin{aligned}
 N_1 &= \frac{1}{8}(1 - \xi)(1 - \eta)(1 + \mu) \\
 N_2 &= \frac{1}{8}(1 - \xi)(1 - \eta)(1 - \mu) \\
 N_3 &= \frac{1}{8}(1 - \xi)(1 + \eta)(1 - \mu) \\
 N_4 &= \frac{1}{8}(1 - \xi)(1 + \eta)(1 + \mu) \\
 N_5 &= \frac{1}{8}(1 + \xi)(1 - \eta)(1 + \mu) \\
 N_6 &= \frac{1}{8}(1 + \xi)(1 - \eta)(1 - \mu) \\
 N_7 &= \frac{1}{8}(1 + \xi)(1 + \eta)(1 - \mu) \\
 N_8 &= \frac{1}{8}(1 + \xi)(1 + \eta)(1 + \mu)
 \end{aligned}$$

(2.60)

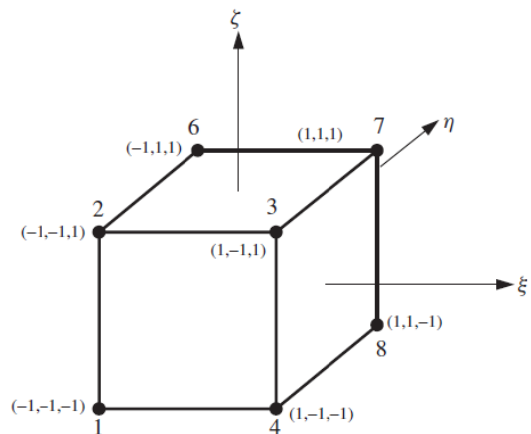


Figure 12 : Fonction de forme cube 8 nœuds

Avec x, y, z

$$\begin{aligned} x &= N_1x_1 + N_2x_2 + N_3x_3 + N_4x_4 + N_5x_5 + N_6x_6 + N_7x_7 + N_8x_8 \\ y &= N_1y_1 + N_2y_2 + N_3y_3 + N_4y_4 + N_5y_5 + N_6y_6 + N_7y_7 + N_8y_8 \\ z &= N_1z_1 + N_2z_2 + N_3z_3 + N_4z_4 + N_5z_5 + N_6z_6 + N_7z_7 + N_8z_8 \end{aligned} \quad (2.61)$$

la matrice B est donnée comme suit

$$[B] = [L] [N] \quad (2.62)$$

Avec

$$[N] = \begin{bmatrix} [N_0] & 0 & 0 \\ 0 & [N_0] & 0 \\ 0 & 0 & [N_0] \end{bmatrix} \quad (2.63)$$

Ou

$$[N_0] = [N_1 \ N_2 \ N_3 \ N_4 \ N_5 \ N_6 \ N_7 \ N_8] \quad (2.64)$$

$$[L]^T = \begin{bmatrix} \frac{\partial(\)}{\partial x} & 0 & 0 \\ 0 & \frac{\partial(\)}{\partial y} & 0 \\ 0 & 0 & \frac{\partial(\)}{\partial z} \\ \frac{\partial(\)}{\partial y} & \frac{\partial(\)}{\partial x} & 0 \\ 0 & \frac{\partial(\)}{\partial z} & \frac{\partial(\)}{\partial y} \\ \frac{\partial(\)}{\partial z} & 0 & \frac{\partial(\)}{\partial x} \end{bmatrix} \quad (2.65)$$

La Jacobien est donné comme suit

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \mu} & \frac{\partial y}{\partial \mu} & \frac{\partial z}{\partial \mu} \end{bmatrix} \quad (2.66)$$

Les dérivées partielles sont évaluées comme suit

$$\frac{\partial f}{\partial x} = \frac{1}{|J|} \begin{vmatrix} \frac{\partial f}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial f}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial f}{\partial \mu} & \frac{\partial y}{\partial \mu} & \frac{\partial z}{\partial \mu} \end{vmatrix} \quad \frac{\partial f}{\partial y} = \frac{1}{|J|} \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial f}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial f}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \mu} & \frac{\partial f}{\partial \mu} & \frac{\partial z}{\partial \mu} \end{vmatrix} \quad \frac{\partial f}{\partial z} = \frac{1}{|J|} \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial f}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial f}{\partial \eta} \\ \frac{\partial x}{\partial \mu} & \frac{\partial y}{\partial \mu} & \frac{\partial f}{\partial \mu} \end{vmatrix} \quad (2.67)$$

la matrice [D] est donnée par

$$[D] = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \quad (2.68)$$

La matrice de rigidité est écrite en termes de triple intégrale comme suit:

$$[k] = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 [B]^T [D] [B] |J| d\xi d\eta d\mu \quad (2.69)$$

Il est clair que l'élément brique linéaire à 8 nœuds a vingt-quatre degrés de liberté - trois à chaque nœud. Par conséquent pour une structure à n nœuds, la matrice de rigidité globale K sera de taille 3n × 3n

Une fois la matrice de rigidité globale K obtenue, nous avons l'équation suivante à résoudre.

$$[K]\{U\} = \{F\} \quad (2.70)$$

Enfin une fois les déplacements et réactions inconnus trouvés, le vecteur de contraintes est obtenu pour chaque élément comme suit:

$$\{\sigma\} = [D][B] \{u\} \quad (2.71)$$

où σ est le vecteur de contrainte dans l'élément, de taille 6×1 et u est le vecteur de déplacement de l'élément, de taille 24×1 . Le vecteur σ s'écrit pour chaque élément comme

$$\{\sigma\} = [\sigma_x \ \sigma_y \ \sigma_z \ \tau_{xy} \ \tau_{yz} \ \tau_{zx}]^T. \quad (2.72)$$

Enfin, le volume de l'élément est donné par la formule suivante:

$$V = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 |J| \, d\xi \, d\eta \, d\mu \quad (2.73)$$

2.11.3 Élément hexaèdre a 20 nœuds

L'élément brique hexaèdre a 20 nœuds est un élément fini tridimensionnel caractérisé par des fonctions de forme quadrilatérales dans chacune des directions x, y et z. Il est également appelé hexaèdron.

L'élément brique quadratique a vingt nœuds avec trois degrés de liberté à chaque nœud, Les coordonnées globales des 20 nœuds sont notées $(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4), (x_5, y_5, z_5), (x_6, y_6, z_6), (x_7, y_7, z_7), (x_8, y_8, z_8), (x_9, y_9, z_9), (x_{10}, y_{10}, z_{10}), (x_{11}, y_{11}, z_{11}), (x_{12}, y_{12}, z_{12}), (x_{13}, y_{13}, z_{13}), (x_{14}, y_{14}, z_{14}), (x_{15}, y_{15}, z_{15}), (x_{16}, y_{16}, z_{16}), (x_{17}, y_{17}, z_{17}), (x_{18}, y_{18}, z_{18}), (x_{19}, y_{19}, z_{19}), (x_{20}, y_{20}, z_{20})$. [Ref \[1\]](#)

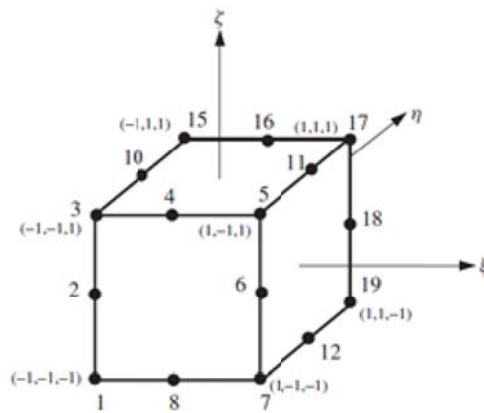


Figure 13 : Hexaedron a 20 nœuds

Pour l'élément quadratique à 20 nœuds, les fonctions de forme peuvent être écrites sous la forme suivante:

$$[N] = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & \dots \\ 0 & N_1 & 0 & 0 & N_2 & 0 & \dots \\ 0 & 0 & N_1 & 0 & 0 & N_2 & \dots \end{bmatrix} \text{ Jusqu'à } N_{20} \quad (2.74)$$

$$\begin{aligned}
 N_1 &= \frac{1}{8}(1 - \xi)(1 - \eta)(1 - \zeta)(-\xi - \eta - \zeta - 2) \\
 N_2 &= \frac{1}{4}(1 - \xi)(1 - \eta)(1 - \zeta^2) \\
 N_3 &= \frac{1}{8}(1 - \xi)(1 - \eta)(1 + \zeta)(-\xi - \eta + \zeta - 2) \\
 N_4 &= \frac{1}{4}(1 - \xi^2)(1 - \eta)(1 + \zeta) \\
 N_5 &= \frac{1}{8}(1 + \xi)(1 - \eta)(1 + \zeta)(\xi - \eta + \zeta - 2) \\
 N_6 &= \frac{1}{4}(1 + \xi)(1 - \eta)(1 - \zeta^2) \\
 N_7 &= \frac{1}{8}(1 + \xi)(1 - \eta)(1 - \zeta)(\xi - \eta - \zeta - 2) \\
 N_8 &= \frac{1}{4}(1 - \xi^2)(1 - \eta)(1 - \zeta) \\
 N_9 &= \frac{1}{4}(1 - \xi)(1 - \eta^2)(1 - \zeta) \\
 N_{10} &= \frac{1}{4}(1 - \xi)(1 - \eta^2)(1 + \zeta)
 \end{aligned}$$

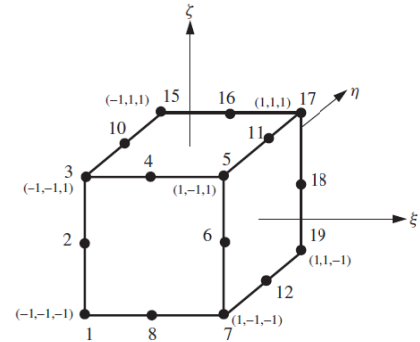


Figure 14 : fonction de forme Hexaedron a 20 nœuds

$$\begin{aligned}
 N_{11} &= \frac{1}{4}(1 + \xi)(1 - \eta^2)(1 + \zeta) \\
 N_{12} &= \frac{1}{4}(1 + \xi)(1 - \eta^2)(1 - \zeta) \\
 N_{13} &= \frac{1}{8}(1 - \xi)(1 + \eta)(1 - \zeta)(-\xi + \eta - \zeta - 2) \\
 N_{14} &= \frac{1}{4}(1 - \xi)(1 + \eta)(1 - \zeta^2) \\
 N_{15} &= \frac{1}{8}(1 - \xi)(1 + \eta)(1 + \zeta)(-\xi + \eta + \zeta - 2) \\
 N_{16} &= \frac{1}{4}(1 - \xi^2)(1 + \eta)(1 + \zeta) \\
 N_{17} &= \frac{1}{8}(1 + \xi)(1 + \eta)(1 + \zeta)(\xi + \eta + \zeta - 2) \\
 N_{18} &= \frac{1}{4}(1 + \xi)(1 + \eta)(1 - \zeta^2) \\
 N_{19} &= \frac{1}{8}(1 + \xi)(1 + \eta)(1 - \zeta)(\xi + \eta - \zeta - 2) \\
 N_{20} &= \frac{1}{4}(1 - \xi^2)(1 + \eta)(1 - \zeta)
 \end{aligned} \tag{2.75}$$

Le vecteur de déformation contient six composants différents du tenseur de déformation:

$$\{\varepsilon\} = \{\varepsilon_x \ \varepsilon_y \ \varepsilon_z \ \gamma_{xy} \ \gamma_{yz} \ \gamma_{zx}\} \tag{2.76}$$

La matrice déformation-déplacement [B] pour les éléments tridimensionnels a l'aspect suivant

$$[B_i] = \begin{bmatrix} \partial N_i/\partial x & 0 & 0 \\ 0 & \partial N_i/\partial y & 0 \\ 0 & 0 & \partial N_i/\partial z \\ \partial N_i/\partial y & \partial N_i/\partial x & 0 \\ 0 & \partial N_i/\partial z & \partial N_i/\partial y \\ \partial N_i/\partial z & 0 & \partial N_i/\partial x \end{bmatrix} \quad (2.77)$$

Les dérivées des fonctions de forme par rapport aux coordonnées globales sont obtenues comme suit

$$\begin{Bmatrix} \partial N_i/\partial x \\ \partial N_i/\partial y \\ \partial N_i/\partial z \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \partial N_i/\partial \xi \\ \partial N_i/\partial \eta \\ \partial N_i/\partial \zeta \end{Bmatrix} \quad (2.78)$$

où la matrice jacobienne a l'apparence:

$$[J] = \begin{bmatrix} \partial x/\partial \xi & \partial y/\partial \xi & \partial z/\partial \xi \\ \partial x/\partial \eta & \partial y/\partial \eta & \partial z/\partial \eta \\ \partial x/\partial \zeta & \partial y/\partial \zeta & \partial z/\partial \zeta \end{bmatrix} \quad (2.79)$$

Les dérivées partielles de x, y, z par rapport à ξ η ζ sont trouvées par différenciation des déplacements exprimé par des fonctions de forme et des valeurs de déplacement nodal:

$$\begin{aligned} \frac{\partial x}{\partial \xi} &= \sum \frac{\partial N_i}{\partial \xi} x_i, & \frac{\partial x}{\partial \eta} &= \sum \frac{\partial N_i}{\partial \eta} x_i, & \frac{\partial x}{\partial \zeta} &= \sum \frac{\partial N_i}{\partial \zeta} x_i \\ \frac{\partial y}{\partial \xi} &= \sum \frac{\partial N_i}{\partial \xi} y_i, & \frac{\partial y}{\partial \eta} &= \sum \frac{\partial N_i}{\partial \eta} y_i, & \frac{\partial y}{\partial \zeta} &= \sum \frac{\partial N_i}{\partial \zeta} y_i \\ \frac{\partial z}{\partial \xi} &= \sum \frac{\partial N_i}{\partial \xi} z_i, & \frac{\partial z}{\partial \eta} &= \sum \frac{\partial N_i}{\partial \eta} z_i, & \frac{\partial z}{\partial \zeta} &= \sum \frac{\partial N_i}{\partial \zeta} z_i \end{aligned} \quad (2.80)$$

La matrice de rigidité est écrite en termes de triple intégrale comme suit:

$$[k] = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 [B]^T [D] [B] |J| d\xi d\eta d\mu \quad (2.81)$$

avec $dV = dx dy dz = |J| d\xi d\eta d\zeta \quad (2.82)$

2.12 Vecteur forces

Les forces agissant sur un élément sont de type

- Charges de Volumes
- Charges surfaciques
- Charges ponctuelles ou nodales
- Ou autres (thermiques etc..)

2.13 Conditions aux limites :

Avant de résoudre le système des équations d'équilibre il faut prendre en compte les conditions aux limites l'une des méthodes se résume comme suit :

- Si on a un déplacement bloqué au nœuds i dans le sens x , y , z ou x et y ou x et z ou y et z on multiplie la rigidité dans la matrice globale $[K]$ à l'adresse correspondante sur la diagonale au nœud i par un chiffre suffisamment grand pour augmenter au maximum la rigidité de ce nœud.
- Si on a un chargement réparti sur une facette d'un élément il sera pris en compte sur le vecteur charge globale, on l'évalue et on injecte la valeur calculée à l'adresse appropriée sur $\{F\}$.

2.14 Méthode de factorisation de Cholesky

Notre choix pour la résolution, s'est arrêté sur la méthode directe de factorisation de Cholesky sur matrice bande symétrique.

Théorème

Factorisation de **Cholesky** d'une matrice :

Si la matrice [A] est symétrique définie positive, il existe au moins une matrice réelle triangulaire inférieure L telle que :

$$A = LL^T$$

On peut également imposer que les éléments diagonaux de la matrice [L] soient tous positifs, et la factorisation correspondante est alors unique.

Algorithme

On cherche la matrice :

$$L = \begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix}$$

on déduit que :

$$a_{ij} = (LL^T)_{ij} = \sum_{k=1}^n l_{ik}l_{jk} = \sum_{k=1}^{\min\{i,j\}} l_{ik}l_{jk}, ; 1 \leq i, j \leq n$$

puisque $l_{ij} = 0$ si $1 \leq i < j \leq n$.

La matrice A étant symétrique, il suffit que les relations ci-dessus soient vérifiées pour $i \leq j$, c'est-à-dire que les éléments l_{ij} de la matrice L doivent satisfaire :

$$a_{ij} = \sum_{k=1}^i l_{ik}l_{jk}, ; 1 \leq i, j \leq n$$

Pour $j=1$, on détermine la première colonne de L :

- ▶ (i=1) $a_{11} = l_{11}l_{11}$ d'où $l_{11} = \sqrt{a_{11}}$
- ▶ (i=2) $a_{12} = l_{11}l_{21}$ d'où $l_{21} = \frac{a_{12}}{l_{11}}$
- ▶ ...
- ▶ (i=n) $a_{1n} = l_{11}l_{n1}$ d'où $l_{n1} = \frac{a_{1n}}{l_{11}}$

puis on détermine la j-ème colonne de L, après avoir calculé les (j-1) premières colonnes :

$$\begin{aligned}
 \blacktriangleright (i=j) \quad a_{ii} &= l_{i1}l_{i1} + \dots + l_{ii}l_{ii} \text{ d'oà } l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2} \\
 \blacktriangleright (i=j+1) \quad a_{i,i+1} &= l_{i1}l_{i+1,1} + \dots + l_{ii}l_{i+1,i} \text{ d'oà } l_{i+1,i} = \frac{a_{i,i+1} - \sum_{k=1}^{i-1} l_{ik}l_{i+1,k}}{l_{ii}} \\
 \blacktriangleright \dots & \\
 \blacktriangleright (i=n) \quad a_{i,n} &= l_{i1}l_{n1} + \dots + l_{ii}l_{ni} \text{ d'oà } l_{ni} = \frac{a_{in} - \sum_{k=1}^{i-1} l_{ik}l_{nk}}{l_{ii}}
 \end{aligned}$$

Résolution de système

Pour la résolution de système linéaire de la forme : $Ax=b$, le système devient

$$LL^T x = b \Leftrightarrow \begin{cases} Ly = b & (1), \\ L^T x = y & (2). \end{cases}$$

On résout le système (1) pour trouver le vecteur y, puis le système (2) pour trouver le vecteur x. La résolution est facilitée par la forme triangulaire des matrices.

2.15 Intégration numérique

Lors de l'évaluation de la matrice de rigidité, des vecteurs forces, et poids propre; on a à calculer des intégration sur le volume et les surfaces .

Or les intégrations sur le volume et la surface de l'élément sont complexes et ne peuvent être calculée manuellement.

L'intégration numérique est essentielle pour évaluation pratique de ces intégrales sur le domaine de l'élément. La méthode d'intégration courante est la méthode d'intégration de Gauss car elle utilise un nombre minimal de points d'échantillonnage pour atteindre le niveau de précision souhaité.

donc Le moyen le plus évident d'obtenir l'intégrale

$$I = \int_{-1}^1 \int_{-1}^1 f(\xi, \eta) d\xi d\eta \quad (2.83)$$

est d'abord d'évaluer l'intégrale interne en gardant η constant, c'est-à-dire,

$$\int_{-1}^1 f(\xi, \eta) d\xi = \sum_{j=1}^n f(\xi_j, \eta) w_j = \psi(\eta) \quad (2.84)$$

En évaluant l'intégrale externe de la même manière, nous avons

$$\begin{aligned} I &= \int_{-1}^1 \psi(\eta) d\eta = \sum_{i=1}^n \psi(\eta_i) w_i \\ &= \sum_{i=1}^n w_i \sum_{j=1}^n f(\xi_j, \eta_i) w_j \\ &= \sum_{i=1}^n \sum_{j=1}^n f(\xi_j, \eta_i) w_i w_j \end{aligned} \quad (2.85)$$

Pour un élément brique 3D de la même manière

$$I = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 f(\xi, \eta, \zeta) d\xi d\eta d\zeta \quad (2.86)$$

$$= \sum_{k=1}^n \sum_{j=1}^n \sum_{i=1}^n f(\xi_i, \eta_j, \zeta_k) w_i w_j w_k \quad (2.87)$$

Le nombre de points d'intégration dans chaque direction a été supposé être le même, ainsi lors de l'intégration numérique, nous désignerons la sommation comme une somme unique sur des points uniques

d'ou pour deux dimensions

$$I = \int_{-1}^1 \int_{-1}^1 f(\xi, \eta) d\xi d\eta = \sum_{l=1}^m f(\xi_l, \eta_l) W_l \quad (2.88)$$

et pour trois dimensions.

$$I = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 f(\xi, \eta, \zeta) d\xi d\eta d\zeta = \sum_{l=1}^m f(\xi_l, \eta_l, \zeta_l) W_l \quad (2.89)$$

le poids W_i désigne le produit des poids unidimensionnels appropriés.

$$I = \int_{-1}^1 \int_{-1}^1 f(\xi, \eta) d\xi d\eta = \sum_{l=1}^m f(\xi_l, \eta_l) W_l \quad (2.90)$$

Élément triangulaires ou tétraédriques

Pour un triangle, en termes de coordonnées d'aire, les intégrales sont de la forme

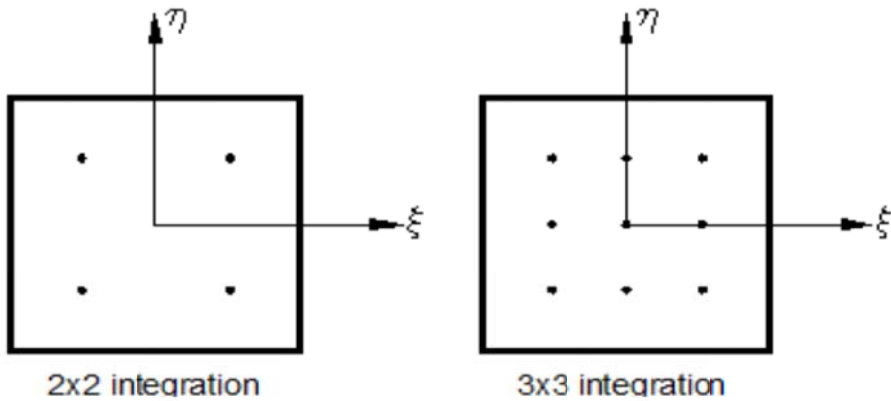
$$I = \int_0^1 \int_0^{1-L_1} f(L_1, L_2, L_3) dL_2 dL_1 \quad (2.91)$$

$$L_3 = 1 - L_1 - L_2 \quad (2.92)$$

Éléments rectangulaires

Le choix de l'ordre d'intégration dépend de la précision souhaitée et du type d'élément sélectionné pour l'analyse.

Coordonnées des points de Gauss et poids dans les éléments rectangulaires

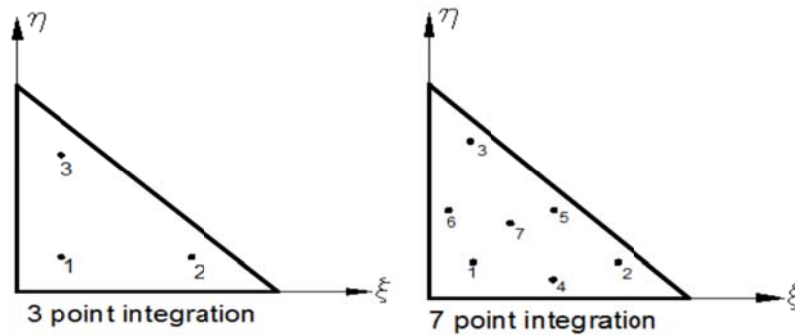


r	ξ	W
1	0.000000 000000 000000	2.000000 000000 000000
2	0.577350 269189 635764	1.000000 000000 000000
3	0.774596 669241 483377	0.555555 555555 555555
	0.000000 000000 000000	0.888888 888888 888888
4	0.861136 371594 052575	0.347854 845137 453857
	0.339981 043584 856264	0.652145 154862 546142
5	0.906179 845938 663992	0.236926 885056 189087
	0.538469 310105 683091	0.478628 670499 366468
	0.000000 000000 000000	0.568888 888888 888888

Tableau 1 : Coordonnées des points de Gauss et poids dans les éléments rectangulaires

Éléments triangulaires

Coordonnées des points de Gauss et poids dans les éléments triangulaires [Ref\[8\]](#)



ordre Integration	ξ_i	η_i	ω_i
3-point	0.1 666 666 666	0.1 666 666 666	0.1 666 666 666
	0.6 666 666 666	0.1 666 666 666	0.1 666 666 666
	0.1 666 666 666	0.6 666 666 666	0.1 666 666 666
7-point	0.1 012 865 073	0.1 012 865 073	0.0 629 695 903
	0.7 974 269 853	0.1 012 865 073	0.0 629 695 903
	0.1 012 865 073	0.7 974 269 853	0.0 629 695 903
	0.4 701 420 641	0.0 597 158 717	0.0 661 970 764
	0.4 701 420 641	0.4 701 420 641	0.0 661 970 764
	0.0 597 158 717	0.4 701 420 641	0.0 661 970 764
	0.3 333 333 333	0.3 333 333 333	0.1125

Tableau 2 : Coordonnées des points de Gauss et poids dans les éléments triangulaires

Pour éléments 3D Tétraèdre

l'intégration Gauss pour l'éléments tétraèdre est données comme suit [Ref\[8\]](#)

ordre Integration	ξ_i	η_i	ζ_i	ω_i
1-point	0.25	0.25	0.25	1
	0.58 541 020	0.13 819 660	0.13 819 660	0.25
	0.13 819 660	0.58 541 020	0.13 819 660	0.25
	0.13 819 660	0.13 819 660	0.58 541 020	0.25
4-point	0.13 819 660	0.13 819 660	0.13 819 660	0.25
	0.13 819 660	0.13 819 660	0.13 819 660	0.25
	0.25	0.25	0.25	-0.8
	1/3	1/6	1/6	0.45
	1/6	1/3	1/6	0.45
5-point	1/6	1/6	1/3	0.45
	1/6	1/6	1/6	0.45

Tableau 3 : Intégration de Gauss pour l'éléments tétraèdre

Pour éléments 3D cubiques Hexaèdre

Caractéristiques de l'intégration avec 8 points de Gauss est données comme suit

Caractéristiques Nœuds	ξ_i	η_i	ζ_i	ω_i
1	$-1/\sqrt{3}$	$-1/\sqrt{3}$	$-1/\sqrt{3}$	1
2	$1/\sqrt{3}$	$-1/\sqrt{3}$	$-1/\sqrt{3}$	1
3	$-1/\sqrt{3}$	$1/\sqrt{3}$	$-1/\sqrt{3}$	1
4	$1/\sqrt{3}$	$1/\sqrt{3}$	$-1/\sqrt{3}$	1
5	$-1/\sqrt{3}$	$-1/\sqrt{3}$	$1/\sqrt{3}$	1
6	$1/\sqrt{3}$	$-1/\sqrt{3}$	$1/\sqrt{3}$	1
7	$-1/\sqrt{3}$	$1/\sqrt{3}$	$1/\sqrt{3}$	1
8	$1/\sqrt{3}$	$1/\sqrt{3}$	$1/\sqrt{3}$	1

Tableau 4 : Intégration avec 8 points de Gauss pour Hexaèdre

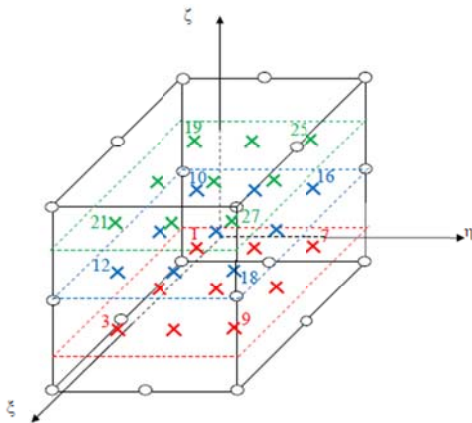


Figure 15 : Intégration 27 points de Gauss Hexaèdre

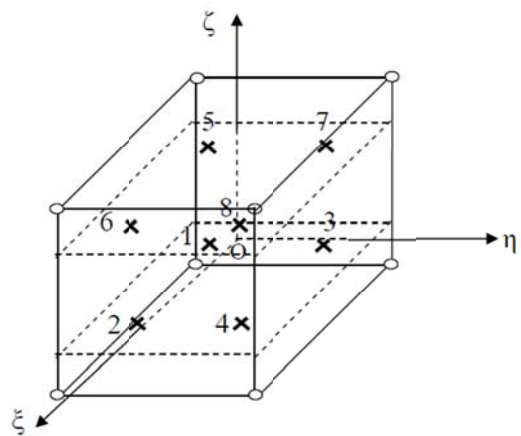


Figure 16 : Intégration 8 points de Gauss Hexaèdre

Caractéristiques Nœuds	ξ_i	η_i	ζ_i	ω_i
1	$-\sqrt{3/5}$	$-\sqrt{3/5}$	$-\sqrt{3/5}$	0,17146776
2	0	$-\sqrt{3/5}$	$-\sqrt{3/5}$	0,27434842
3	$\sqrt{3/5}$	$-\sqrt{3/5}$	$-\sqrt{3/5}$	0,17146776
4	$-\sqrt{3/5}$	0	$-\sqrt{3/5}$	0,27434842
5	0	0	$-\sqrt{3/5}$	0,43895748
6	$\sqrt{3/5}$	0	$-\sqrt{3/5}$	0,27434842
7	$-\sqrt{3/5}$	$\sqrt{3/5}$	$-\sqrt{3/5}$	0,17146776
8	0	$\sqrt{3/5}$	$-\sqrt{3/5}$	0,27434842
9	$\sqrt{3/5}$	$\sqrt{3/5}$	$-\sqrt{3/5}$	0,17146776
10	$-\sqrt{3/5}$	$-\sqrt{3/5}$	0	0,27434842
11	0	$-\sqrt{3/5}$	0	0,43895748
12	$\sqrt{3/5}$	$-\sqrt{3/5}$	0	0,27434842
13	$-\sqrt{3/5}$	0	0	0,43895748
14	0	0	0	0,70233196
15	$\sqrt{3/5}$	0	0	0,43895748
16	$-\sqrt{3/5}$	$\sqrt{3/5}$	0	0,27434842
17	0	$\sqrt{3/5}$	0	0,43895748
18	$\sqrt{3/5}$	$\sqrt{3/5}$	0	0,27434842
19	$-\sqrt{3/5}$	$-\sqrt{3/5}$	$\sqrt{3/5}$	0,17146776
20	0	$-\sqrt{3/5}$	$\sqrt{3/5}$	0,27434842
21	$\sqrt{3/5}$	$-\sqrt{3/5}$	$\sqrt{3/5}$	0,17146776
22	$-\sqrt{3/5}$	0	$\sqrt{3/5}$	0,27434842
23	0	0	$\sqrt{3/5}$	0,43895748
24	$\sqrt{3/5}$	0	$\sqrt{3/5}$	0,27434842
25	$-\sqrt{3/5}$	$\sqrt{3/5}$	$\sqrt{3/5}$	0,17146776
26	0	$\sqrt{3/5}$	$\sqrt{3/5}$	0,27434842
27	$\sqrt{3/5}$	$\sqrt{3/5}$	$\sqrt{3/5}$	0,17146776

Tableau 5 : Intégration avec 27 points de Gauss pour Hexaèdre

2.16 Les méthodes de résolution des équations d'équilibre

Il y a deux stratégies fondamentales de résolution en dynamique, la méthode de superposition modale et la méthode de résolution directe.

Le choix entre ces 02 stratégies dépend de la nature du problème (linéaire ou non linéaire) et du contenu fréquentiel de l'excitation.

La méthode de résolution directe est utilisée pour les problèmes non linéaires ou si le contenu fréquentiel de l'excitation est susceptible d'exciter un grand nombre de mode de la structure, ce qui est le cas dans la propagation d'ondes, l'analyse directe d'une structure en régime transitoire implique l'intégration pas à pas des équations du mouvement.

$$[M].\ddot{u} + [C].\dot{u} + [K].u = \{R(t)\} \quad (2.93)$$

La méthode de superposition modale convient aux structures linéaires dont les premiers modes sont susceptibles d'être excités. [Ref \[12\]](#)

2.16.1 Méthode de superposition modale

En présence d'une structure ayant n degrés de libertés, donc n possibilités de se mouvoir, la méthode de superposition modale se propose de résoudre l'équation de mouvement de cette structure soumise à une excitation dynamique.

La méthode suit le schéma suivant :

Etape 1 : On détermine les matrices de rigidités et de masse [M], [C] et [K]

Etape 2 : On détermine la solution dans sa base propre, on arrive donc à passer d'une fonction complexe $\{u_i\}$ à une série de fonctions connues $\{\phi_i\}$

Tel que :

$$\{u_i\} = Y_i(t) \cdot \{\phi_i\} \quad (2.94)$$

$$\{u(t)\} = \sum_i Y_i(t) \cdot \{\phi_i\} \quad (2.95)$$

Etape 3 : On injecte les expressions de $u(t)$, $\dot{u}(t)$, $\ddot{u}(t)$ Ecrites en fonction des $\{\varphi_i\}$ dans l'équation de mouvement qui devient

$$[M].\sum_i Y_i(t) . \{\varphi_i\} + [C].\sum_i Y_i(t) . \{\varphi_i\} + [K].\sum_i Y_i(t) . \{\varphi_i\} = \{R(t)\} \quad (2.96)$$

Etape 4 : a partir de l'orthogonalité par rapport aux matrices de masse, de rigidité et d'amortissement, et qui est la base de la méthode de superposition modale

$$\begin{cases} \{\phi_i\}^T [M] \{\phi_j\} = 1; i = j \\ \{\phi_i\}^T [M] \{\phi_j\} = 0; i \neq j \\ \{\phi_i\}^T [K] \{\phi_j\} = 1; i = j \\ \{\phi_i\}^T [K] \{\phi_j\} = 0; i \neq j \end{cases} \quad (2.97)$$

$$\{\phi_i\}^T [C] \{\phi_j\} = 2\omega\delta_{ij}$$

Tel que δ_{ij} symbole de Kronecker

On multiplie l'équation (2.96) par $\{\phi_i\}^T$ vecteur transposé.

$$\{\phi_i\}^T [M] \sum_j Y_j(t) \{\phi_j\} + \{\phi_i\}^T [C] \sum_j Y_j(t) \{\phi_j\} + \{\phi_i\}^T [K] \sum_j Y_j(t) \{\phi_j\} = \{\phi_i\}^T \{R(t)\} \quad (2.98)$$

L'exploitation du résultat de l'orthogonalité ramène l'équation (2.98) sous forme :

$$\{\phi_i\}^T [M] Y_i(t) \{\phi_i\} + \{\phi_i\}^T [C] Y_i(t) \{\phi_i\} + \{\phi_i\}^T [K] Y_i(t) \{\phi_i\} = \{\phi_i\}^T \{R(t)\} \quad (2.99)$$

On pose

$$\{\phi_i\}^T \{R(t)\} = R_i * (t) \quad (\text{Force excitatrice généralisée du mode } i)$$

$$\{\phi_i\}^T [M] \{\phi_i\} = M_i * \quad (\text{Masse généralisée du mode } i)$$

$$\{\phi_i\}^T [K] \{\phi_i\} = K_i * \quad (\text{Rigidité généralisée du mode } i)$$

De plus

$$\omega_i^2 = \frac{K_i *}{M_i *} \quad (2.100)$$

$$\{\phi_i\}^T [C] \{\phi_i\} = 2\omega \quad (2.101)$$

En exploitant les égalités citées ci-dessus (2.100) et (2.101) et en divisant l'équation (2.99) par $M_i *$, on aboutit à une équation de la forme :

$$\ddot{y}_i(t) + 2\omega_i \dot{y}_i(t) + \omega_i^2 y_i(t) = \frac{R_i * (t)}{M_i *} \quad (2.102)$$

On arrive ainsi à passer d'un système couple à N degré de liberté vers N systèmes équivalents entre eux à une inconnue $Y_i(t)$.

Cette opération s'appelle un découplage de mode, elle n'a été possible que grâce à l'orthogonalité des modes.

La solution de l'équation (2. 102) est donnée par l'intégral de Duhamel :

$$Y_i(t) = \frac{1}{M_i \cdot \omega_i} \int_0^t P_i^*(t) \cdot \exp\left[-\int_0^t \omega_i(t-s)\right] \sin \omega_i(t-s) ds \quad (2.103)$$

N.B. : Ce cheminement nous montre que pour obtenir la solution, il faut déterminer les pulsations ω_i et vecteur $\{\varphi_i\}$ De chaque mode. [Ref \[13\]](#)

Pour déterminer les pulsations ω_i et vecteurs $\{\varphi_i\}$ On considère un système non amorti, avec des oscillations libres ; L'équation de mouvement s'écrit alors :

$$[M].\ddot{u} + [K].u = 0 \quad (2.104)$$

La solution de l'équation (4. 10) est donnée par :

$$u = \{\varphi\} \sin \omega (t - t_0) \quad (2.105)$$

On injectant l'expression (4. 11) dans (4. 10) on obtient

$$[K]\{\varphi\} = \omega^2 \cdot [M]\{\varphi\} \quad (2.106)$$

Pour chaque mode l'équation (4. 12) s'écrit

$$[K]\{\varphi_i\} = \omega_i^2 \cdot [M]\{\varphi_i\} \quad (2.107)$$

Où ω_i^2 : Valeur du mode i

$\{\varphi_i\}$: Vecteur propre du mode i

Etape 5 : passage aux coordonnées géométriques :

(Déplacement du mode i)

$$\{u_i\} = Y_i \cdot \{\varphi_i\} \quad (2.108)$$

Etape 6 : calcul le déplacement total

$$\{u(t)\} = \sum_i \{u_i\} \quad (2.109)$$

Etape 7 : On calcul les forces élastiques

$$\{F_k\} = [K] \cdot \{u(t)\} \quad (2.110)$$

2.16.2 Intégration pas à pas

L'opération de base lors de la résolution pas à pas sur les équations différentielles de mouvement (2.93) est leur conversion en un ensemble d'équation algébrique.

On y parvient en introduisant une relation simple entre déplacement, vitesse et accélération, relation que suppose valable pendant un court incrément de temps.

On peut ainsi exprimer les évolutions incrémentales de vitesse et de déplacement en fonction des évolutions de l'accélération, ou encore les évolutions de vitesse et d'accélération en fonction des déplacements incrémentaux.

Dans l'un ou l'autre cas, seul un vecteur d'inconnus subsiste dans les équations incrémentales de l'équilibre, et ainsi résolution d'équations simultanées se fera par n'importe quelle méthode standard.

La conversion fera passer l'équilibre (2.93) à la forme suivante :

$$\hat{K}\{u\} = \hat{R}(t) \quad (2.111)$$

Tels que : $\hat{K}(t) = f([K], [M], [C]) \quad (2.112)$

$$\hat{R}(t) = f([K], [M], [C]) \quad (2.113)$$

2.16.3 Les méthodes d'intégrations directe

La philosophie adoptée par les différentes méthodes d'intégration est la même, ils adoptent une hypothèse permettant d'introduire une relation simple entre déplacement, vitesse et accélération, toutefois, la différence réside dans la forme de cette hypothèse, il suit qu'une grande variété d'expression pourrait être employée, cependant le schéma de solution doit être effectif, donc on ne considère qu'un nombre limité de schémas qui sont.

- Méthode des différences Centrées
- Méthode de Houbolt
- Méthode de Wilson θ
- **Méthode de Newmark**

2.16.4 Method de Newmark

Considérée comme une extension de la méthode de l'accélération linéaire les hypothèses suivantes sont utilisées [Ref \[12\]](#)

$$\dot{u}^{t+\Delta t} = \dot{u}^t + [(1-\gamma)\ddot{u}^t + \gamma\ddot{u}^{t+\Delta t}]\Delta t \quad (2.114)$$

$$u^{t+\Delta t} = u^t + \dot{u}^t\Delta t + \left[\frac{1}{2} - \beta\right]\ddot{u}^t + \beta\ddot{u}^{t+\Delta t}\Delta t^2 \quad (2.115)$$

Où : γ et β sont des paramètres déterminés pour obtenir une bonne stabilité ainsi qu'une bonne précision.

Quand $\gamma = \frac{1}{2}$ et $\beta = \frac{1}{2}$ les relations (2.114) et (2.115) correspondent à la méthode de l'intégration linéaire.

Après transformation on obtient :

$$\hat{K} u^{t+\theta\Delta t} = \hat{R}^{t+\theta\Delta t} \quad (2.116)$$

avec

$$\hat{K} = K + a_0 M + a_1 C \quad (2.117)$$

$$\hat{R}^{t+\theta\Delta t} = \hat{R}^{t+\theta\Delta t} + M(a_0 u^t + a_2 \dot{u}^t + a_3 \ddot{u}^t) + C(a_1 u^t + a_4 \dot{u}^t + a_5 \ddot{u}^t) \quad (2.118)$$

Où :

$$a_0 = \frac{1}{\beta\Delta t^2} \quad ; \quad a_1 = \frac{\gamma}{\beta\Delta t} \quad ; \quad a_2 = \frac{1}{\beta\Delta t}$$

$$a_3 = \frac{1}{2\beta} - 1 \quad ; \quad a_4 = \frac{\gamma}{\beta} - 1 \quad ; \quad a_5 = \frac{\Delta t}{2} \cdot \frac{\gamma}{\beta} - 2 \quad (2.119)$$

$$a_6 = \Delta t(1 - \gamma) \quad ; \quad a_7 = \gamma\Delta t$$

Les accélérations et vitesses au temps $(t + \Delta t)$ sont données par

$$\ddot{u}^{t+\Delta t} = a_0(u^{t+\Delta t} - u^t) - a_2 \dot{u}^t - a_3 \ddot{u}^t \quad (2.120)$$

$$\dot{u}^{t+\Delta t} = \dot{u}^t + a_7 \ddot{u}^{t+\Delta t} + a_6 \ddot{u}^t \quad (2.121)$$

2.16.5 Algorithme de la méthode de Wilson θ

La méthode d'intégration retenue est la méthode de *Wilson* θ pour ces performances de stabilités inconditionnelles et de sa précision.

Calcul des matrices de rigidité [K] de masse [M] et d'amortissement [C]

avec conditions initiales u^0 , \dot{u}^0 et \ddot{u}^0

sélection d'un pas Δt et on calcul les constantes pour $\theta = 1.4$

$$\begin{aligned} a_0 &= \frac{6}{(\theta\Delta t)^2} & ; & \quad a_1 = \frac{3}{\theta\Delta t} & ; & \quad a_2 = 2a_1 \\ a_3 &= \frac{\theta\Delta t}{2} & ; & \quad a_4 = \frac{a_0}{\theta} & ; & \quad a_5 = -\frac{a_2}{\theta} \\ a_6 &= 1 - \frac{3}{\theta} & ; & \quad a_7 = \frac{\Delta t}{2} & ; & \quad a_8 = \frac{\Delta t^2}{6} \end{aligned} \quad (2.122)$$

matrice de rigidité effective :

$$\hat{K} = K + a_0 M + a_1 C \quad (2.123)$$

On la matrice de rigidité [K] effective factorisée par la méthode de Cholesky

$$\hat{K} = L \cdot D \cdot L^T$$

Pour chaque pas :

vecteur charge effective au temps $(t + \Delta t)$

$$\hat{R}^{t+\theta\Delta t} = R^t + \theta(R^{t+\Delta t} - R^t) + M(a_0 u^t + a_2 \dot{u}^t + 2 \ddot{u}^t) + C(a_1 u^t + 2 \dot{u}^t + a_3 \ddot{u}^t) \quad (2.124)$$

Résolution du système par la substitution de Cholesky pour obtenir les déplacements au temps $(t + \theta\Delta t)$

$$L \cdot D \cdot L^T u^{t+\theta\Delta t} = \hat{R}^{t+\theta\Delta t} \quad (2.125)$$

calcul des déplacements, vitesses et accélérations au temps $(t + \Delta t)$

$$\begin{aligned} \ddot{u}^{t+\Delta t} &= a_4(u^{t+\Delta t} - u^t) - a_5 \dot{u}^t - a_6 \ddot{u}^t \\ \dot{u}^{t+\Delta t} &= \dot{u}^t + a_7(\ddot{u}^{t+\Delta t} + \ddot{u}^t) \\ u^{t+\theta\Delta t} &= u^t + \Delta t \dot{u}^t + a_8(\ddot{u}^{t+\theta\Delta t} + 2 \ddot{u}^t) \end{aligned} \quad (2.126)$$

Le choix du pas d'intégration constitue une étape très sensible pour le déroulement d'un schéma d'intégration.

D'un côté, il doit être assez petit pour la stabilité et la précision de la solution mais d'un autre cote le pas de temps ne doit pas être plus petit que le nécessaire car cela augmenterait le coût du traitement considérablement.

Les schémas d'intégration se subdivisent en méthodes conditionnellement stables ou le pas est pris nécessairement plus petit qu'un pas de temps critique.

La méthode des différences centrées est une de ces méthodes inconditionnellement stables qui quel que soit le rapport entre l'incrément de temps et la période la plus courte ne diverge pas

L'une des plus simples méthodes est la méthode de *Wilson θ* qui pour $\theta > 1.37$; devient inconditionnellement stable

Pour une bonne précision on prend généralement $\theta = 1.4$

La méthode utilisée dans notre programme est la méthode de Newmark

Chapitre 3

Présentation du Programme

Apl Design MARS

Version 1.0

Sommaire

3.2	Présentation du langage de programmation Matlab	78
3.3	Structure d'un programme d'éléments finis.....	79
3.4	Structures du fichier de données.....	81
3.4.1	Fichier de données pour l'élément barre	82
3.4.2	Fichier de données pour l'éléments Poutre.....	83
3.4.3	Fichier de données pour les éléments 2D.....	84
3.4.4	Fichier de données pour les éléments 3D.....	85
3.5	Génération de maillage.....	86
3.6	Apl_Élément barre	95
3.6.1	Interface du programme	95
3.6.2	Sous programmes.....	99
3.6.3	Organigramme de résolution	100
3.7	Apl Elément Poutre	101
3.7.1	Interface du programme Apl_Poutre.....	101
3.7.2	Sous programmes.....	105
3.7.3	Organigramme de résolution et sousroutines.....	106
3.8	Apl Elément triangulaire T6	108
3.8.1	Interface du programme	108
3.8.2	Sous programmes.....	110
3.8.3	Organigramme de calculs.....	111
3.9	Apl Elément quadratique iso-paramétrique Q8	112
3.9.1	Interface du programme	112
3.9.2	Sous programmes.....	115
3.9.3	Organigramme de résolution	116
3.10	Apl Elément tétraédrique linéaire (T4)	117
3.10.1	Interface du programme	117
3.10.2	Sous programmes.....	119
3.10.3	Organigramme de résolution	120

3.11	Apl Élément brique linéaire (solide C8).....	121
3.11.1	Interface du programme	121
3.11.2	Sous Programmes	123
3.11.3	Organigramme de résolution	125
3.12	Apl Élément hexaédron (C20)	126
3.12.1	Interface du programme	126
3.12.2	Sous programme	128
3.12.3	Organigramme de résolution	129

3.1 Introduction

APL Design_MARS version 1.0 est un programme d'Analyse des Structures par Éléments Finis version 1.0 développé sous l'environnement Matlab

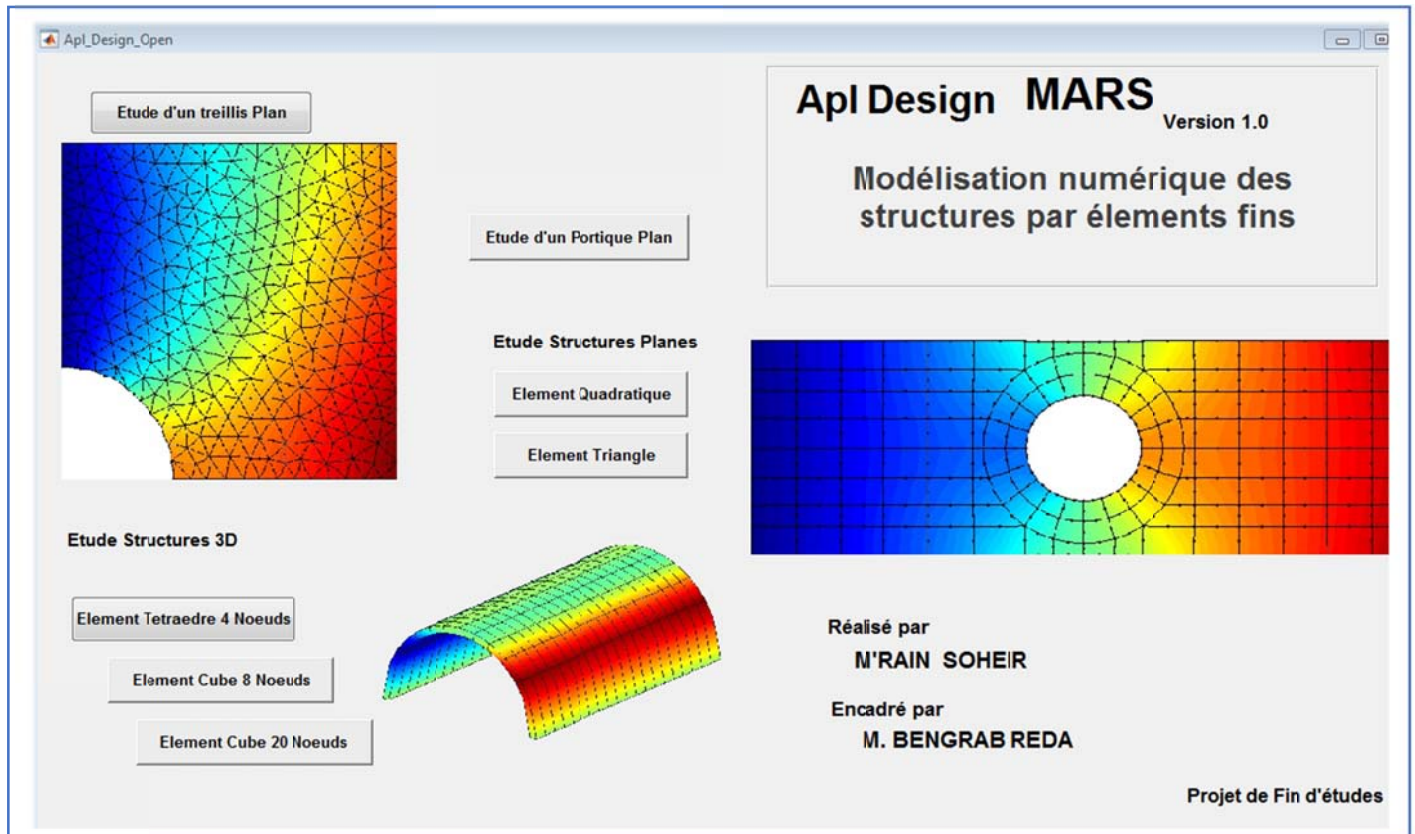


Figure 17 : interface principale du programme *Apl design_Stutures MARS* version 1.0

Caractéristiques générales de **Apl Design Structure MARS** version 1.0

Apl_Design Structure MARS version 1.0 permet d'effectuer une Analyse des Structures par la méthode des éléments finis ; il comprend les modules suivants

- **Apl_Design_Barre** : Élément treillis .
- **Apl_Design_Poutre** : Élément poutre.
- **Apl_Design_Q8** : Élément 2D Quadratique iso-paramétrique à 8 Nœuds .
- **Apl_Design_Tri** : Élément 2D triangulaire iso-paramétrique à 6 Nœuds .
- **Apl_Design_T4** : Élément 3D tétraèdre linéaire à 4 Nœuds.
- **Apl_Design_C8** : Élément 3D hexaèdre à 8 Nœuds.
- **Apl_Design_C20** : Élément 3D hexaèdre à 20 Nœuds.

Types de Structure étudiée : Barres, Poutres, solides 2D et 3D.

Types d'éléments générés par maillage automatique : Elément Quadratique iso-paramétrique à 8 nœuds, triangle à 6 nœuds, tétraèdre à 4 nœuds, hexaèdre à 8 nœuds, hexaèdre à 20 nœuds.

Langage De programmation : Matlab

Apl_Design Structure MARS version 1.0 est utilisée sous l'environnement du logiciel Matlab, ou compilé en cas d'une utilisation isolée.

La structure générale du programme est présentée de manière où chaque module travail séparément avec une interface différente adaptée qui facilite son utilisation au maximum

AVANTAGES DU PROGRAMME :

- ✓ Création des modèles géométriques et maillage de différents types d'éléments quadratiques et linéaires dans le domaine 2D et 3D
- ✓ Analyse avec des éléments quadratiques (éléments quadratiques rectangles à 8 nœuds, éléments hexaédriques à 20 nœuds avec 8 et 27 points de Gauss.
- ✓ Cartographie donnant la répartition des contraintes ou des déformations.
- ✓ Simulation graphique sous chargement progressive pour les différents modules.
- ✓ Note de calcul et visualisation des matrices de rigidité élémentaires et globale A l'écran ou sous forme de fichiers textes.
- ✓ Aperçu à l'écran des valeurs des différents type de contraintes pour chaque élément fini à part aux points de gauss.

3.2 Présentation du langage de programmation Matlab

MATLAB est le nom usuel de **Matrix Laboratory**, est un logiciel commercial écrit en C, développé par MathWorks. Il constitue un environnement puissant, complet et facile à utiliser, destiné au calcul scientifique. Disposant d'un système interactif, intégrant les calculs numériques et les visualisations, Matlab contient plusieurs centaines (voire milliers, selon les versions et les modules optionnels autour du noyau Matlab) de fonctions mathématiques, scientifiques et techniques. L'approche matricielle avec Matlab permet de traiter les données sans aucune limitation de taille et de réaliser des calculs numériques et symboliques d'une façon fiable et rapide.

Ce langage de programmation (Matlab) est utile pour illustrer comment programmer la méthode des éléments finis car il permet de coder très rapidement des méthodes numériques et dispose d'une vaste bibliothèque mathématique prédéfinie, et de nombreux outils d'algèbre linéaire sont déjà définis et que le développeur peut se concentrer entièrement sur l'implémentation de l'algorithme en fonction de la structure des données.

Les fonctions mathématiques et graphiques étendues libèrent davantage le développeur des corvées de développer ces fonctions ainsi que la possibilité de visualiser rapidement les résultats. [Ref \[4\]](#)

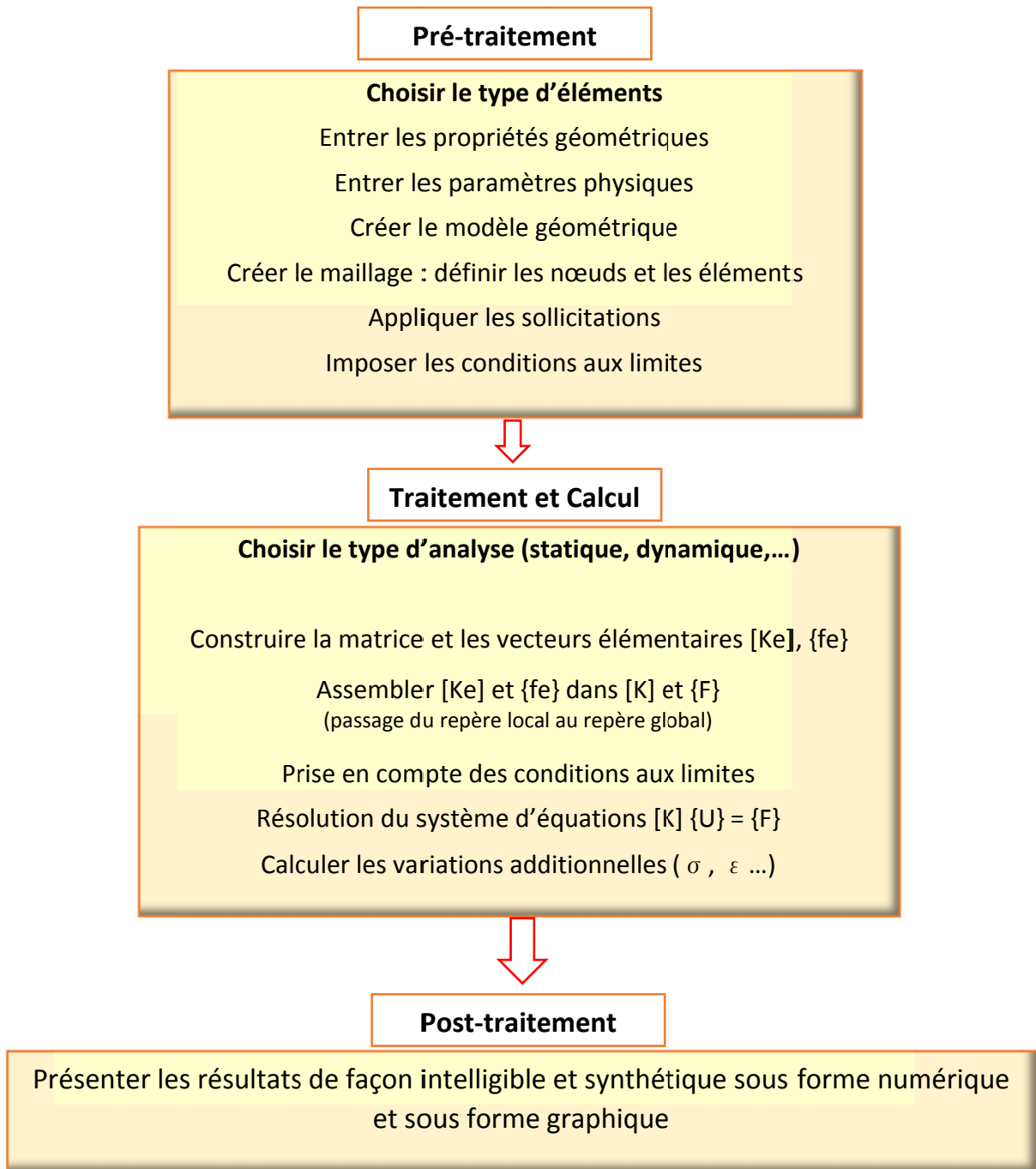
Un simple programme d'élément bidimensionnel dans Matlab n'a besoin que de quelques centaines de lignes de code alors qu'en Fortran ou C ++, il peut en avoir besoin de quelques milliers.

La programmation avec Matlab se fait par bloc (sous-programme) appelé fonction. Elle enregistre les paramètres éventuels (arguments) et renvoie des résultats (valeur de retour) suivant une procédure précise. Les fonctions sont stockées dans un fichier dont le nom se termine par **.m**. Les figures créées par **GUI** Matlab possèdent toujours des fonctions d'exécution correspondantes.

3.3 Structure d'un programme d'éléments finis

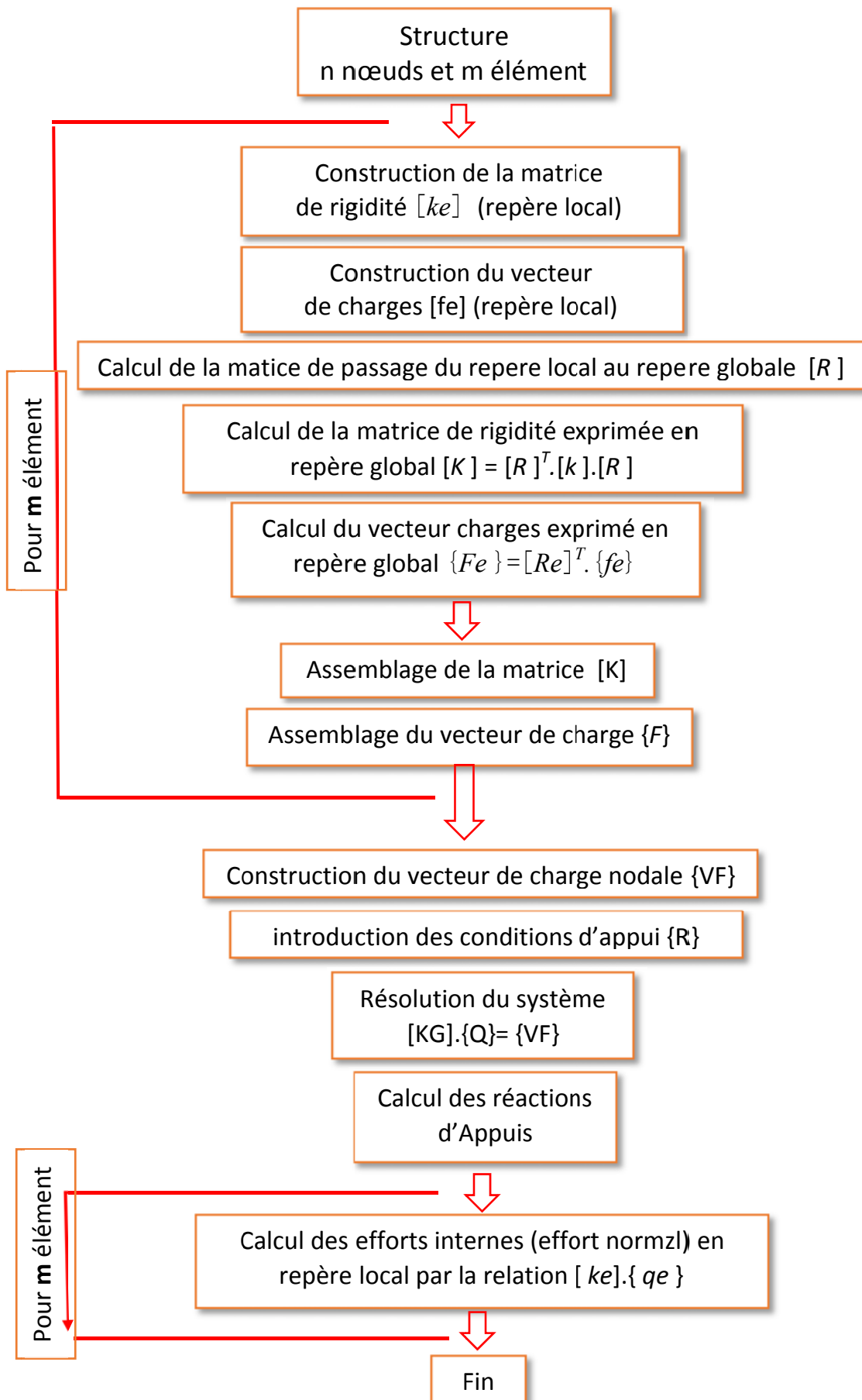
Un programme d'éléments finis typique est structuré comme suit :

1. Pré-traitement
2. Traitement et Calcul
3. Post-traitement



Organigramme 1 : Structure d'un programme d'éléments finis

Organigramme général de résolution



3.4 Structure du fichier de données

Dans la section de prétraitement, les données de la structure sont définies dans un fichier externe aux programmes il s'agit notamment de la discrétisation des éléments, des propriétés des matériaux, les vecteurs de force, les conditions aux limites appliquées.

La structures de données utilisées dans la méthode des éléments finis et en particulier de celles qui sont implémentées dans l'exemple de code sont quelque peu arbitraires dans la mesure où l'on peut imaginer de nombreuses manières de stocker les données d'un programme élément fini, mais nous essayons d'utiliser les structures les plus flexibles et propices à Matlab. La conception de ces structures de données peut dépendre du langage de programmation utilisé, mais elles ne sont généralement pas significativement différentes de celles décrites ici.

Matrice de coordonnées nodales

Puisque nous programmons la méthode des éléments finis, il n'est pas surprenant que nous ayons besoin d'un moyen de représenter la discrétisation des éléments du domaine.

Pour ce faire, nous définissons un ensemble de nœuds et un ensemble d'éléments qui relie ces nœuds d'une manière ou d'une autre. Les coordonnées des nœuds sont stockées dans la matrice de coordonnées nodales.

La dimension de cette matrice est $(nn \text{ } sdim)$ où nn est le nombre de nœuds et $sdim$ est le nombre de dimensions spatiales du problème 2D ou 3D.

Matrice de connectivité des éléments

Les définitions des éléments sont stockées dans la matrice de connectivité des éléments. Il s'agit d'une matrice de numéros de nœuds où chaque ligne de la matrice contient la connectivité d'un élément selon le type de l'élément de référence étudié (éléments barres, poutres, solides linéaire et quadratique 2d et 3d).

Notez que les connectivités des éléments sont toutes ordonnées dans un sens bien déterminer car cela peut entraîner des jacobiens négatifs et donc des matrices d'états singulières et ce qui est évidemment faux !!!

3.4.1 Fichier de données pour l'élément barre

Diagram illustrating the data file structure for a bar element, with callouts identifying key sections:

- Nombre de nœuds**: 7
- Coordonnées num x y**:

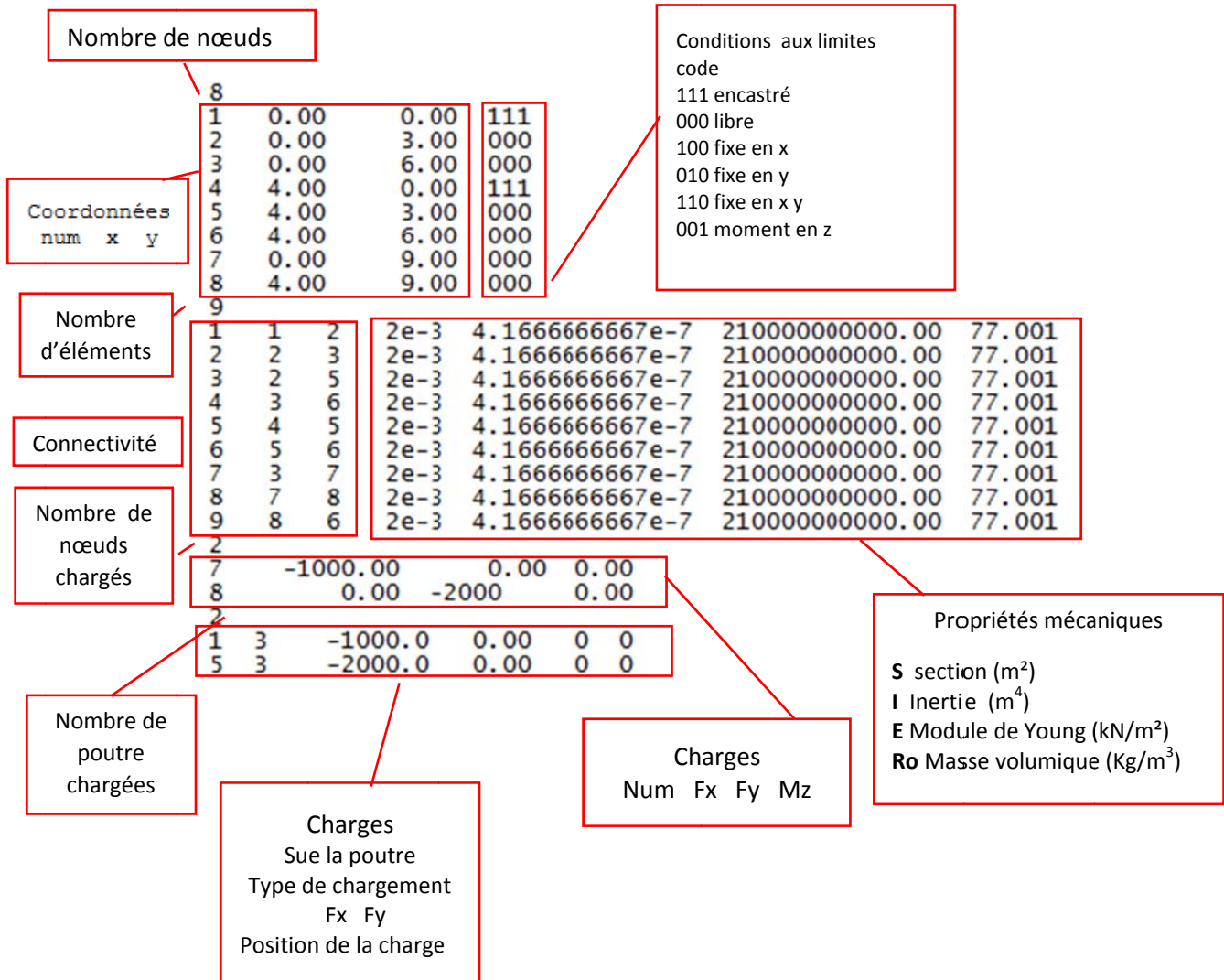
1	0.00000	0.00000
2	2.00000	1.50000
3	4.00000	0.00000
4	6.00000	1.50000
5	8.00000	0.00000
6	10.00000	1.50000
7	12.00000	0.00000
- cod**:


```
11 Encastrement
10 fixe sans x
01 fixe sans y
00 libre
```
- Nombre d'éléments**: 11
- Matrice de Connectivité**:

1	1	3
2	1	2
3	3	2
4	2	4
5	3	4
6	3	5
7	5	4
8	4	6
9	5	6
10	5	7
11	7	6
- Propriétés mécaniques**:
 - a largeur de la section (m)
 - b hauteur de la section (m)
 - E Module de Young (kN/m²)
 - Ro Masse volumique(Kg/m³)
- Charge num Fx Fy**:

2	0.000	-3.000
5	0.000	-6.000
6	8.000	0.000

3.4.2 Fichier de données pour l'éléments Poutre



3.4.3 Fichier de données pour les éléments 2D

Nombre d'éléments

Nombre de Materiaux

Nombre de nœuds

Coordonnées num x y

Propriétés des matériaux
E Module de Young
Ro Masse volumique
v coefficient de Poisson

Connectivite

Nombre de nœuds chargés

Charge num Fx Fy

```

29 6 1
1 1.000 1.000 11
2 1.000 2.500 10
3 1.000 4.000 10
4 1.000 5.500 10
5 1.000 7.000 10
6 1.000 8.500 10
7 1.000 10.000 10
8 2.500 1.000 11
9 2.500 4.000 00
10 2.500 7.000 00
11 2.500 10.000 00
12 4.000 1.000 11
13 4.000 2.500 00
14 4.000 4.000 00
15 4.000 5.500 00
16 4.000 7.000 00
17 4.000 8.500 00
18 4.000 10.000 00
19 5.500 1.000 11
20 5.500 4.000 00
21 5.500 7.000 00
22 5.500 10.000 00
23 7.000 1.000 11
24 7.000 2.500 10
25 7.000 4.000 10
26 7.000 5.500 10
27 7.000 7.000 10
28 7.000 8.500 10
29 7.000 10.000 10

cod
11 Encastrement
10 fixe sans x
01 fixe sans y
00 libre

1 1 12 14 3 8 13 9 2 1 1 1 1.0
2 3 14 16 5 9 15 10 4 1 1 1 1.0
3 5 16 18 7 10 17 11 6 1 1 1 1.0
4 12 23 25 14 19 24 20 13 1 1 1 1.0
5 14 25 27 16 20 26 21 15 1 1 1 1.0
6 16 27 29 18 21 28 22 17 1 1 1 1.0

1 1000000.000 0.00 0.300
3
7 0.000 -500000.0
11 0.000 -200000.0

Charge
num Fx Fy
    
```

3.4.4 Fichier de données pour les éléments 3D

Nombre de nœuds		Nombre d'éléments		Nombre de Matériaux					
24		5		1					

Coordonnées Num XYZ	1	0	0	0	111	Conditions aux limites code 111 encastré 000 libre 100 fixe en x 010 fixe en y 100 fixe en z 110 fixe en x et y 101 fixe en x et z 011 fixe en y et z
	2	0	1	0	111	
	3	0	1	1	111	
	4	0	0	1	111	
	5	1	0	0	0	
	6	1	1	0	0	
	7	1	1	1	0	
	8	1	0	1	0	
	9	2	0	0	0	
	10	2	1	0	0	
	11	2	1	1	0	
	12	2	0	1	0	
	13	3	0	0	0	
	14	3	1	0	0	
	15	3	1	1	0	
	16	3	0	1	0	
	17	4	0	0	0	
	18	4	1	0	0	
	19	4	1	1	0	
	20	4	0	1	0	
	21	5	0	0	0	
	22	5	1	0	0	
	23	5	1	1	0	
	24	5	0	1	0	

Connectivité		Code du matériau							
1	1	5	6	2	4	8	7	3	1
2	5	9	10	6	8	12	11	7	1
3	9	13	14	10	12	16	15	11	1
4	13	17	18	14	16	20	19	15	1
5	17	21	22	18	20	24	23	19	1

Nombre nœuds chargés		Propriétés des matériaux	
1	2300000000.0000	25.0	0.30
2			
23	0.000	0.000	-1.00
24	0.000	0.000	-1.00

Charges	
num	Fx Fy Fz

3.5 Génération de maillage

La génération de maillage avec **Apl_Mesh_guide** permet de générer un maillage en 2D pour plusieurs types d'éléments T3 T6 Q4 Q8 et plaque avec un trou au centre. Un fichier de coordonnées de chaque nœuds de la structure ainsi que la connectivité des éléments est données avec construction du fichier data prêt à l'emploi.

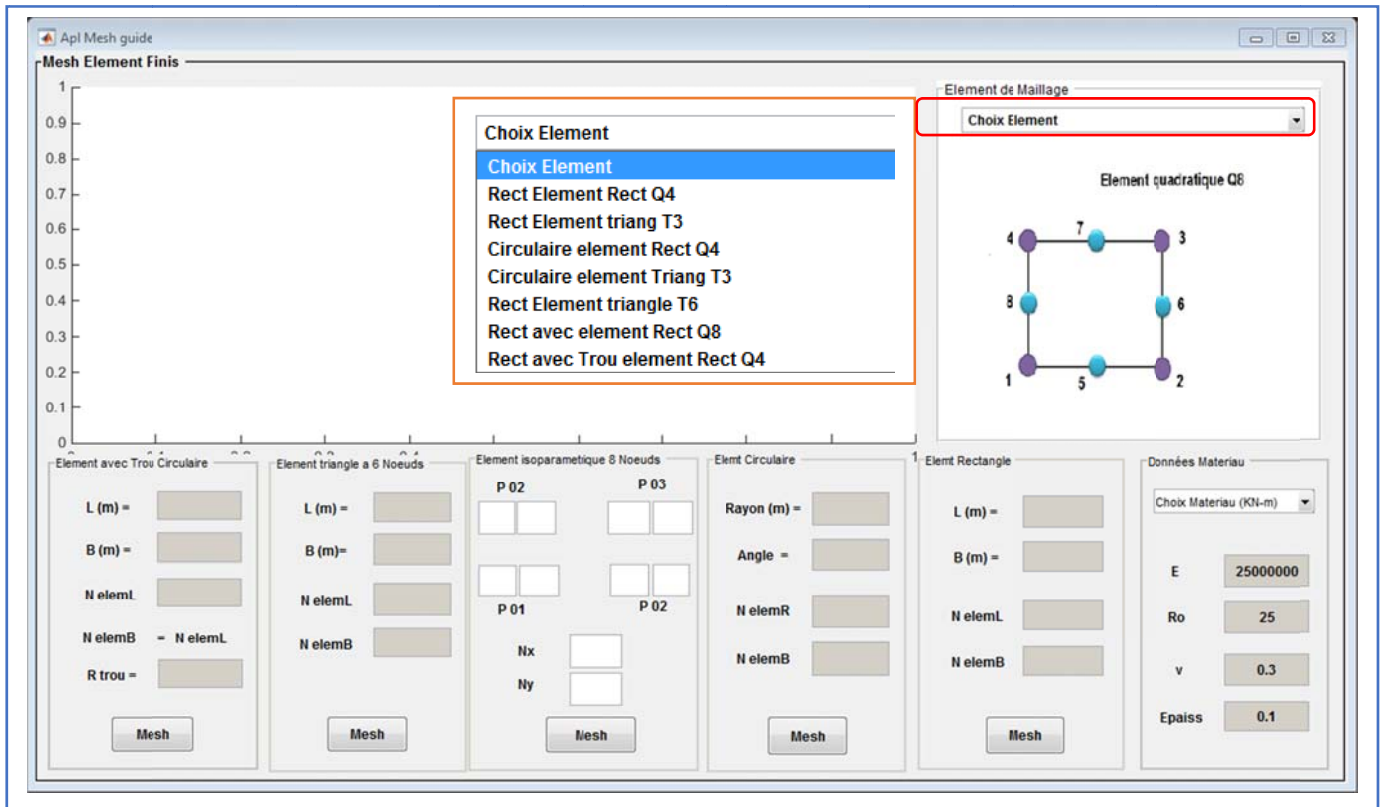


Figure 18 : Interface Générateur de Maillage pour différents élément 2D

Il s'agit de définir en premier lieu le type d'élément avec lequel la structure sera maillée puis la forme et les dimensions à introduire.

- Forme rectangulaire avec l'élément triangle linéaire T3,
- Forme rectangulaire avec l'élément rectangle linéaire Q4,
- Forme rectangulaire avec l'élément iso-paramétrique triangle T6,
- Forme rectangulaire avec l'élément quadratique iso-paramétrique Q8,
- Forme circulaire avec l'élément rectangle linéaire Q4,
- Forme circulaire avec l'élément triangle linéaire T3
- Forme rectangulaire ayant un trou circulaire au centre avec l'élément rectangle linéaire Q4

Après avoir introduit les longueurs géométriques ainsi que le nombre de division dans les sens x et y.

Le maillage est affichée avec les numéros des nœuds et des éléments aussi un fichier de données correspondant au type d'éléments est automatiquement généré de même que les fichiers coordonnés 'CoodXY.dat' et connectivité ; 'connect.dat'.

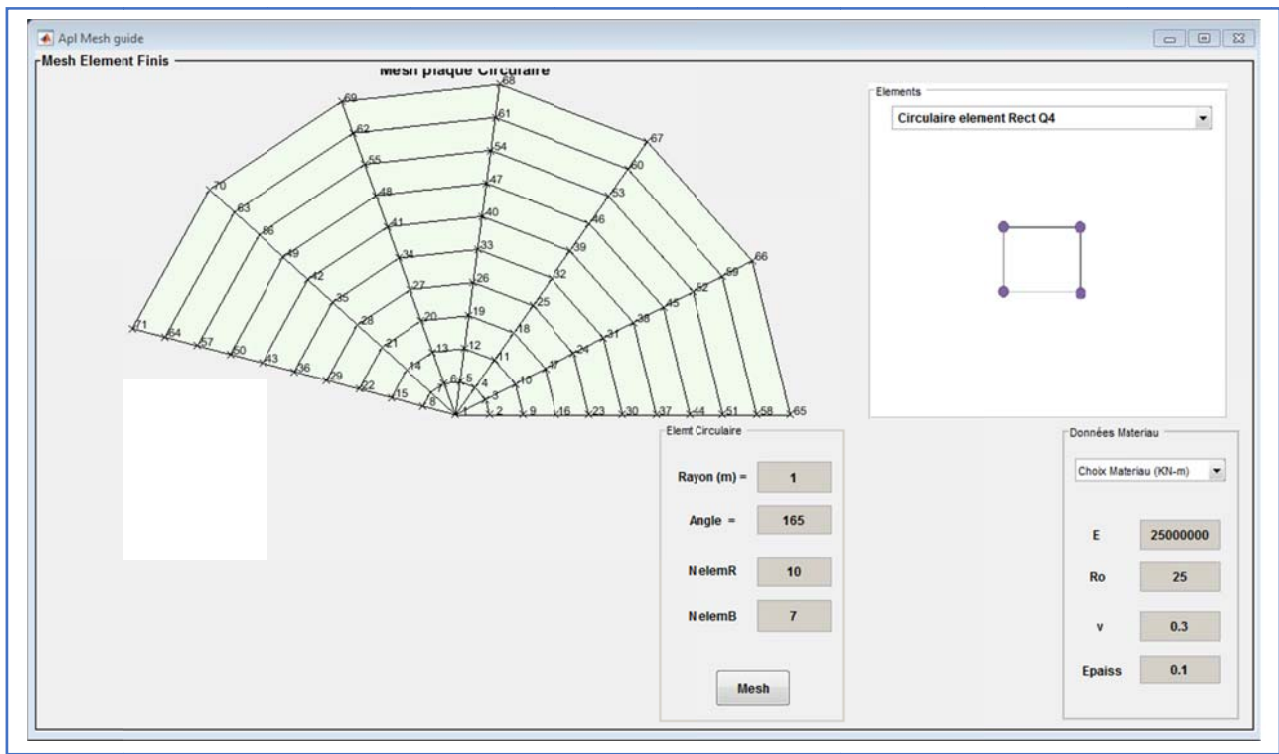


Figure 19 : forme circulaire maillée avec un élément rectangulaire Q4

Ou avec un élément triangle T3.

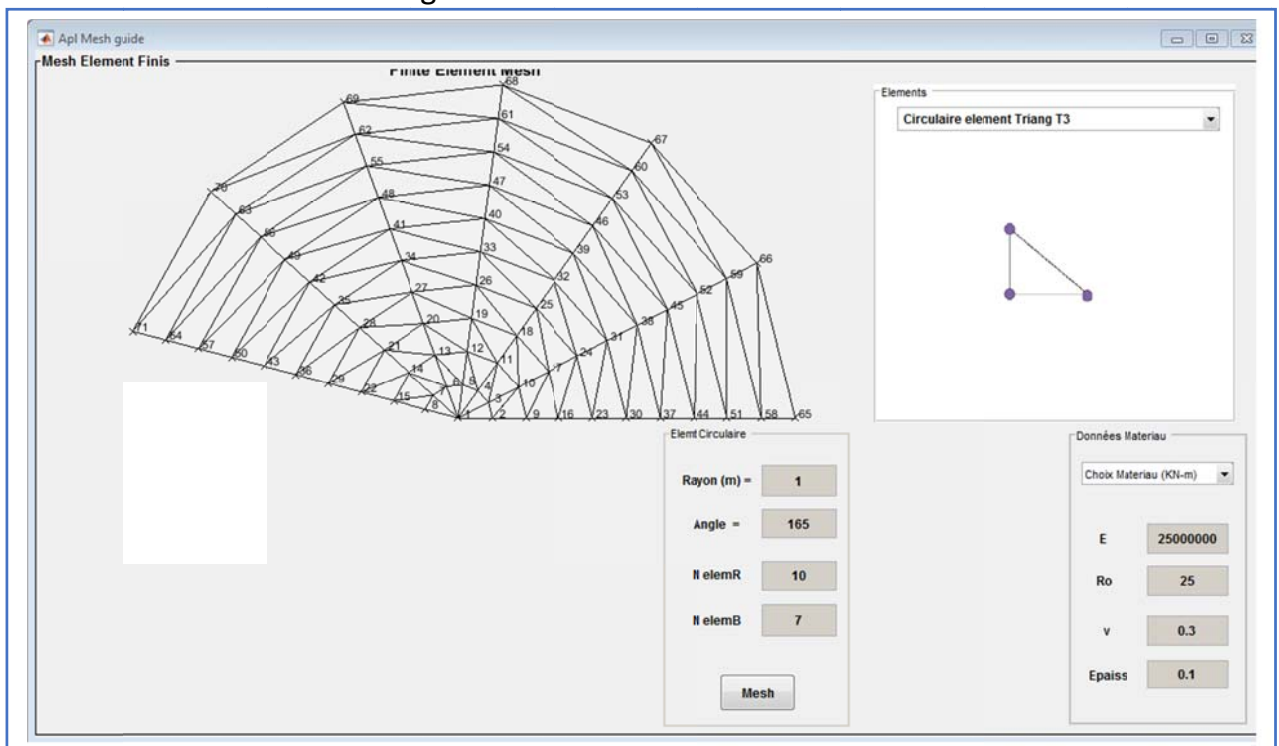


Figure 20 : forme circulaire maillée avec un élément triangle T3

Forme rectangulaire ayant un trou circulaire au centre maillée avec l'élément rectangulaire linéaire Q4.

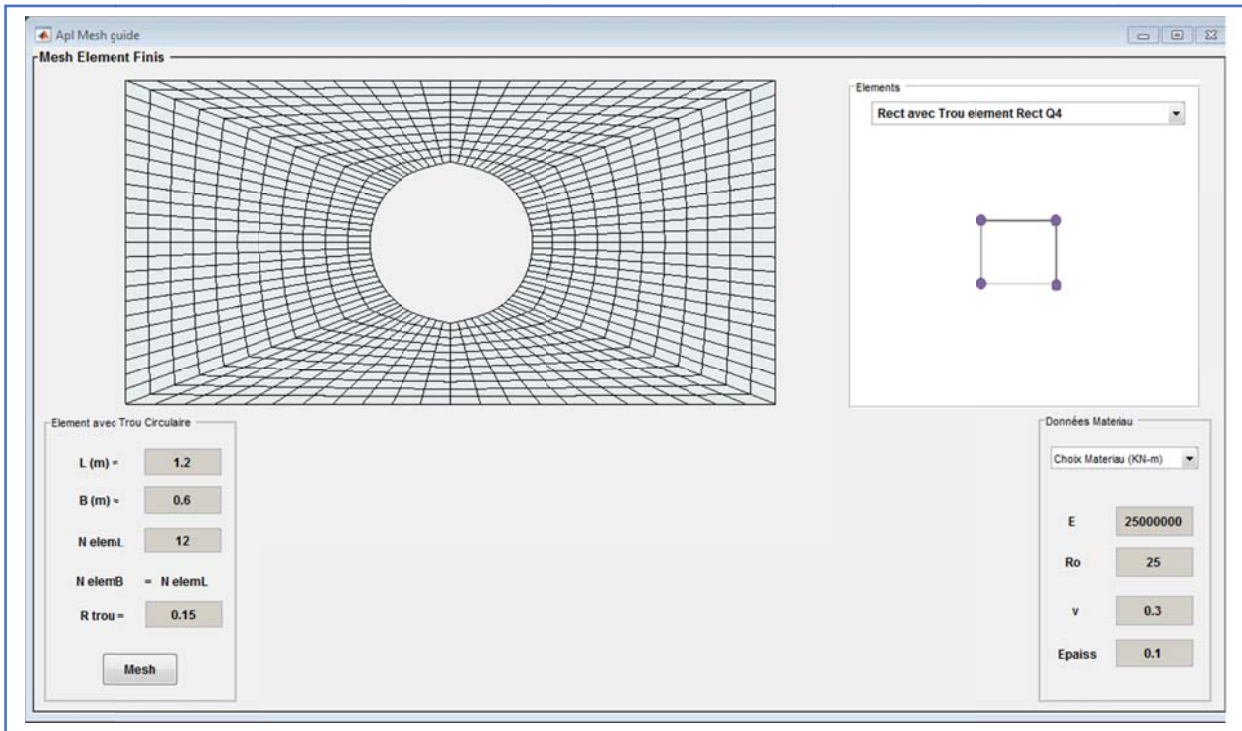


Figure 21 : forme rectangulaire avec trou circulaire au centre

Forme rectangulaire quelconque maillée avec l'élément isoparamétrique quadratique Q8,

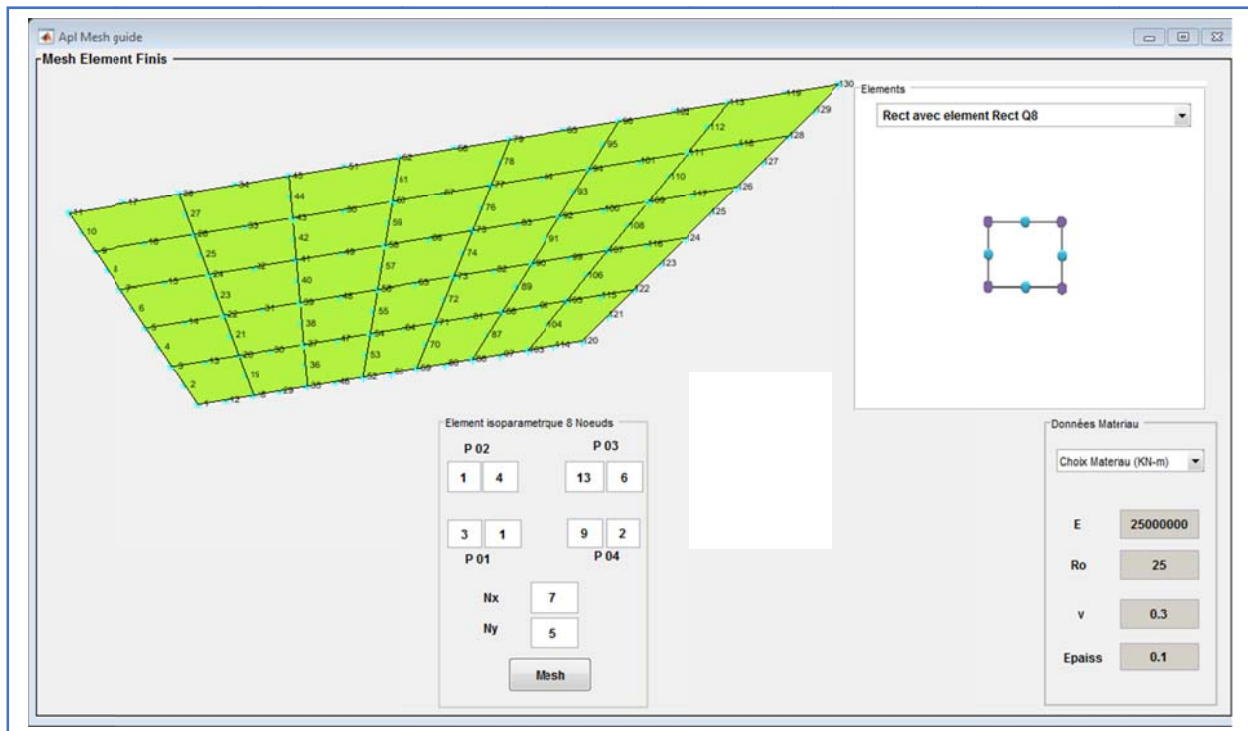


Figure 22 : Élément quadratique iso-paramétrique Q8

Interface générateur de Maillage avec l'élément tétraèdre à 4 nœuds.

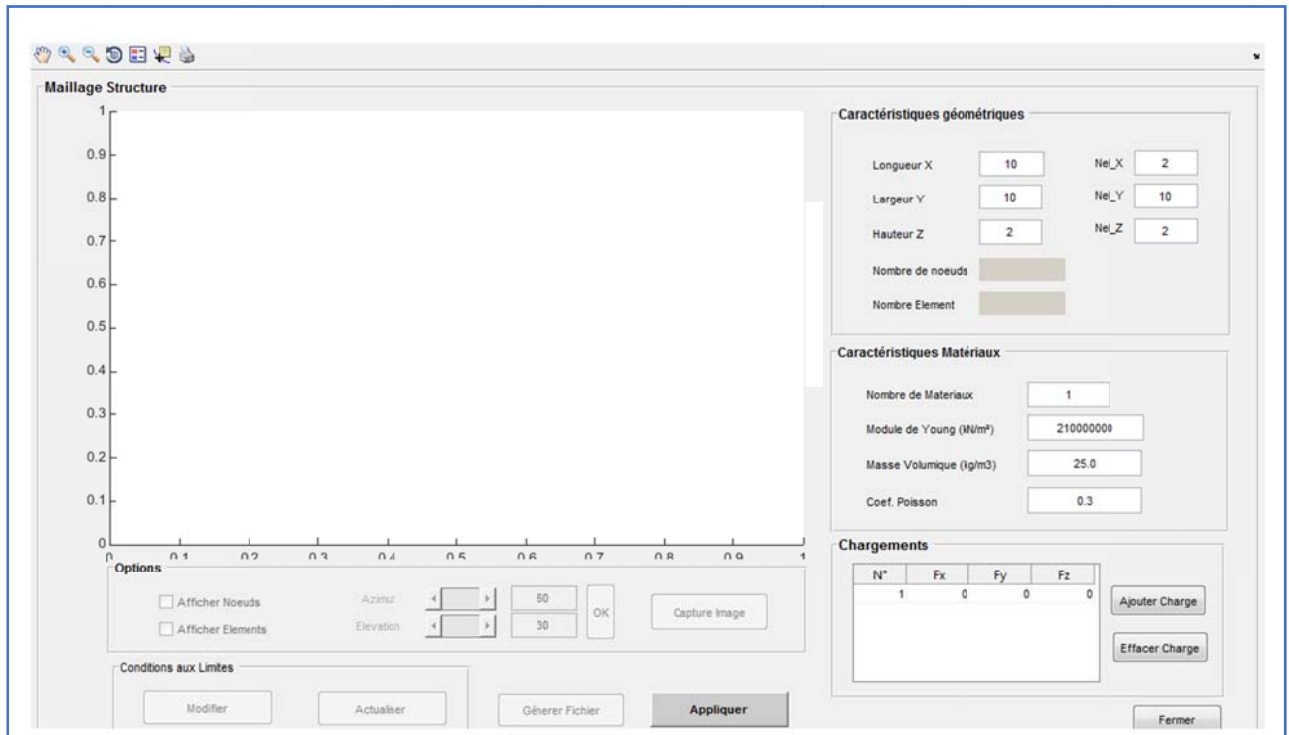


Figure 23 : Interface générateur de Maillage pour élément tétraèdre T4

Ou après avoir entrée les valeurs de longueur, largeur et hauteur de la forme géométrique ainsi que le nombre de division dans les sens x y et z la construction du maillage est généré et la structure est affichée dans l'interface graphique.

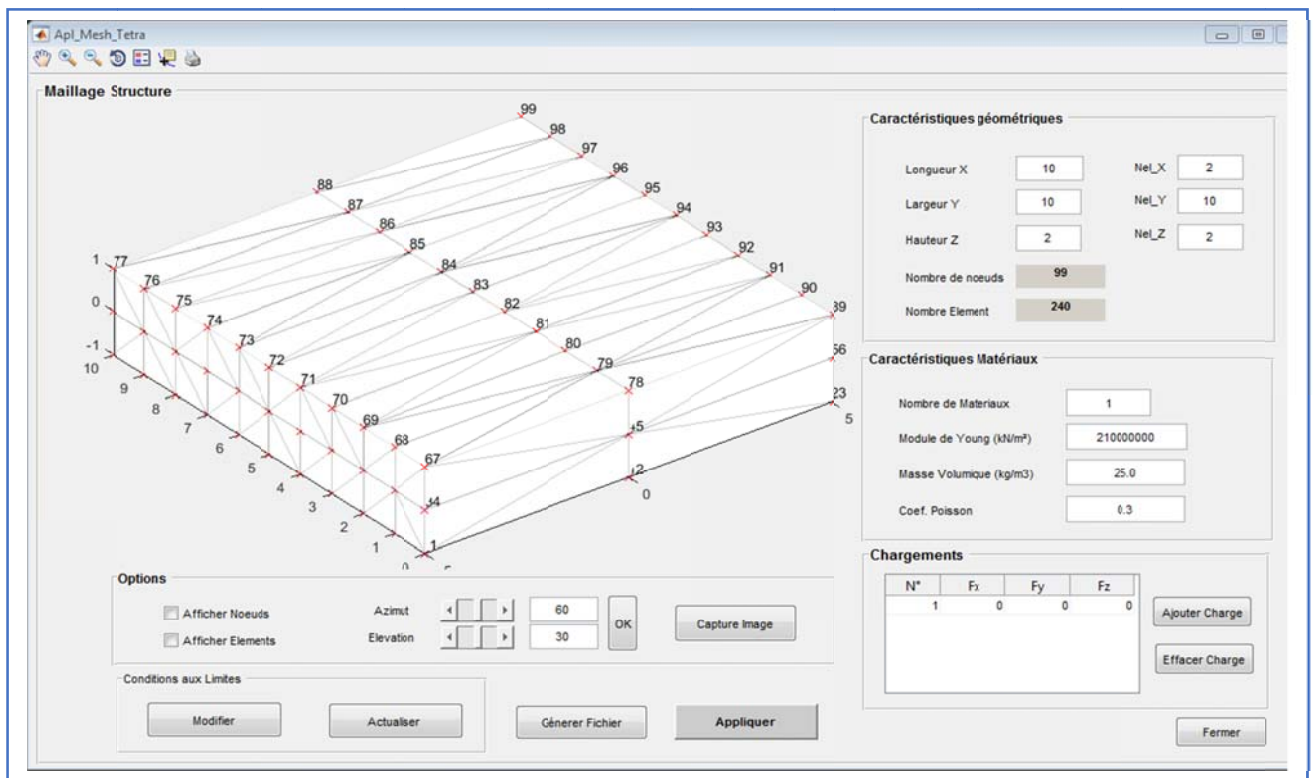
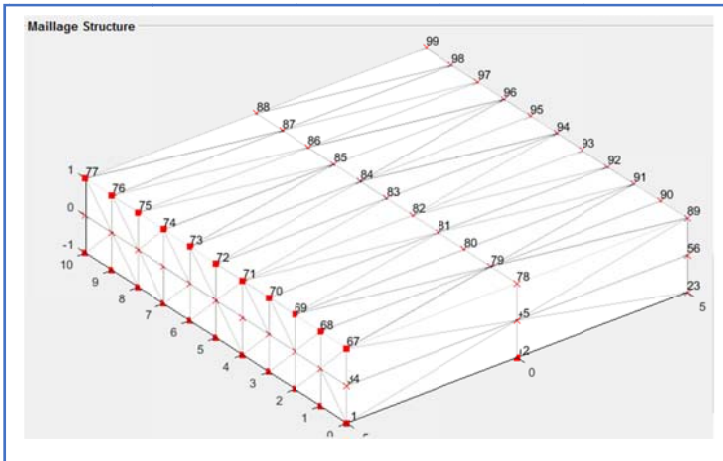


Figure 24 : Exemple Maillage Tétraèdre T4

Il est possible d'introduire les conditions aux limites et de les visualiser en vue d'éventuelle modification.



Fichier data

Coordonnées		Chargement			
elmn	Coord_X	Coord_Y	Coord_Z	Code	
1	-5.00	0.00	-1.00	111	▼
2	-5.00	1.00	-1.00	111	▼
3	-5.00	2.00	-1.00	111	▼
4	-5.00	3.00	-1.00	111	▼
5	-5.00	4.00	-1.00	111	▼
6	-5.00	5.00	-1.00	111	▼
7	-5.00	6.00	-1.00	111	▼
8	-5.00	7.00	-1.00	111	▼
9	-5.00	8.00	-1.00	111	▼
10	-5.00	9.00	-1.00	111	▼
11	-5.00	10.00	-1.00	111	▼
12	0.00	0.00	-1.00	111	▼
13	0.00	1.00	-1.00	111	▼
14	0.00	2.00	-1.00	111	▼
15	0.00	3.00	-1.00	111	▼
16	0.00	4.00	-1.00	111	▼
17	0.00	5.00	-1.00	111	▼
18	0.00	6.00	-1.00	111	▼
19	0.00	7.00	-1.00	111	▼
20	0.00	8.00	-1.00	0	▼
21	0.00	9.00	-1.00	0	▼
22	0.00	10.00	-1.00	0	▼
--	--	--	--	--	▼

- de visualiser les numéros des nœuds et des éléments et de faire une capture de la structure sous format (jpg) ;

Options

Afficher Nosuds Azimut: 60

Afficher Elements Elevation: 30

OK Capture Image

- de des zooms in et out ou un déplacement panoramique Pan ainsi que des rotations dans les trois directions



Après avoir introduit les paramètres du matériaux et les charges extérieures un fichier de données est généré automatiquement sous format fichier.txt ou fichier.dat ou encore fichier.gom ainsi que deux fichiers distincts comportant les coordonnées ainsi que les connectivité coordXYZ.dat et connect.dat.

Caractéristiques Matériaux

Matériau: Beton_E25

Nombre de Matériaux: 1

Module de Young (kN/m²): 25000000

Poids Volumique (kN/m3): 24.5166

Coef. Poisson: 0.2

Chargements

N°	Fx	Fy	Fz
1	0	0	0

Ajouter Charge

Effacer Charge

Interface générateur de maillage avec l'élément Hexaèdre à 8 nœuds.

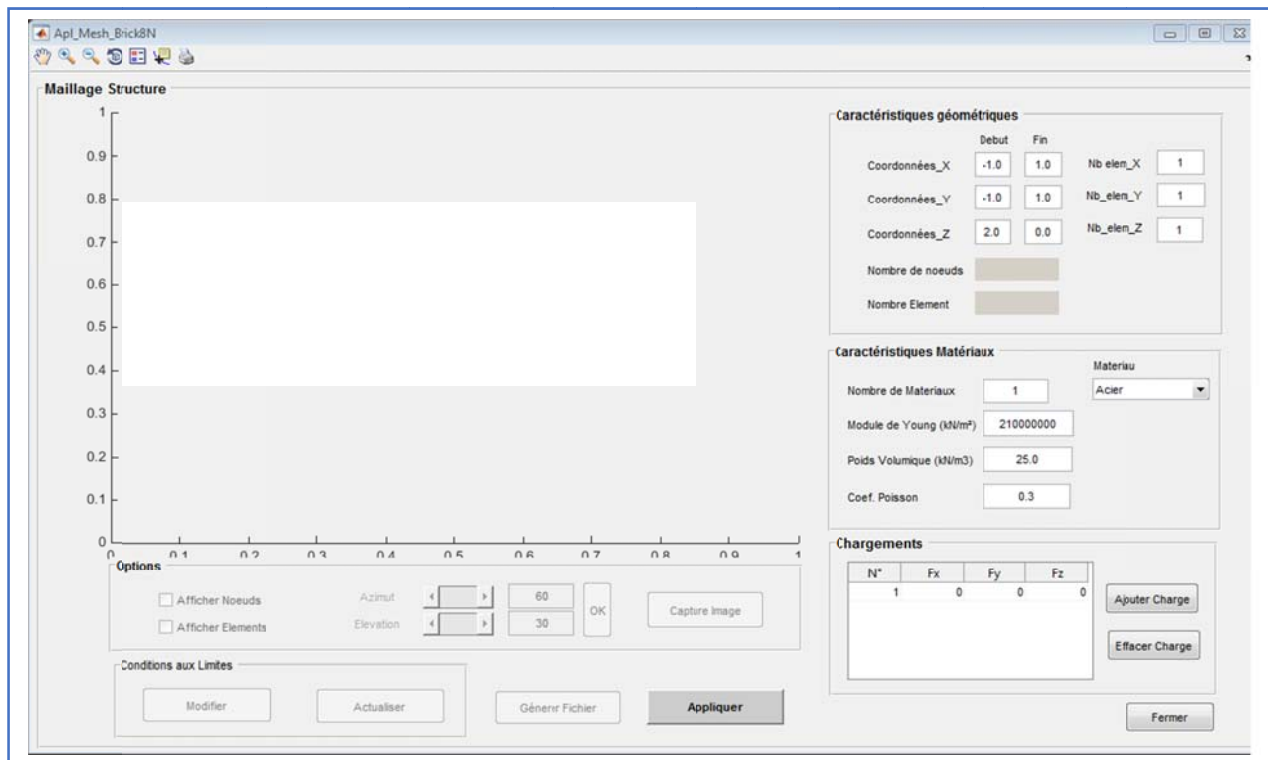


Figure 25 : Interface générateur de Maillage pour élément Hexaèdre a 8 nœuds

Ou après avoir entrée les valeurs de longueur, largeur et hauteur de la forme géométrique ainsi que le nombre de division dans les sens x y et z la construction du maillage est généré et la structure est affichée dans l'interface graphique.

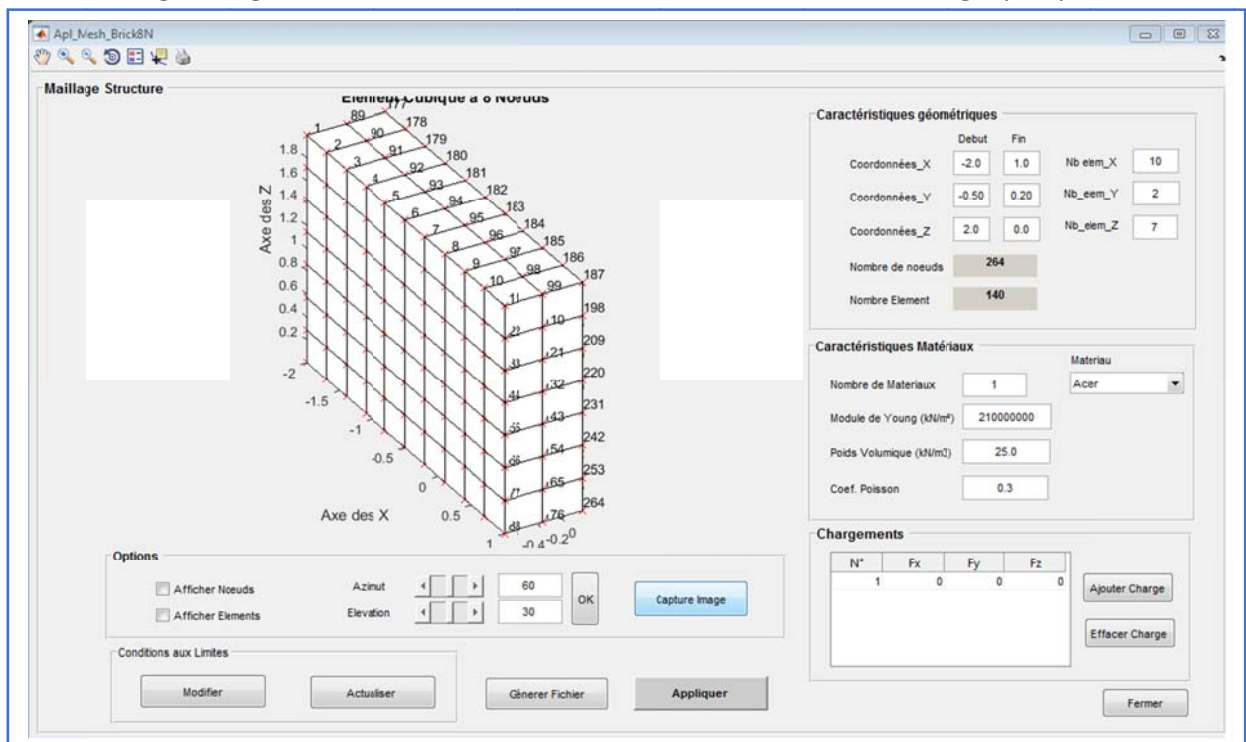
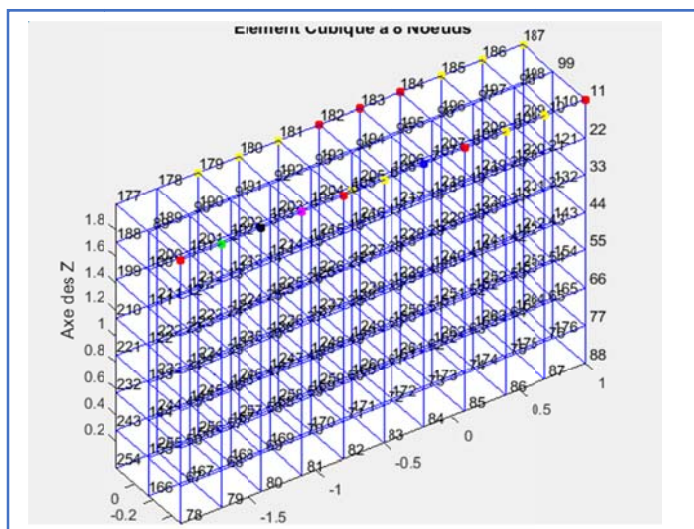


Figure 26 : affichage maillage de la structure

Il est possible d'introduire les conditions aux limites et les visualiser en vu d'éventuelle modification.



Coordonnées		Chargement		
emn	Coord_X	Coord_Y	Coord_Z	Code
1	-2.00	-0.50	2.00	111
2	-1.70	-0.50	2.00	1
3	-1.40	-0.50	2.00	11
4	-1.10	-0.50	2.00	100
5	-0.80	-0.50	2.00	111
6	-0.50	-0.50	2.00	101
7	-0.20	-0.50	2.00	10
8	0.10	-0.50	2.00	111
9	0.40	-0.50	2.00	101
10	0.70	-0.50	2.00	101
11	1.00	-0.50	2.00	111
12	-2.00	-0.50	1.71	0
13	-1.70	-0.50	1.71	0
14	-1.40	-0.50	1.71	0
15	-1.10	-0.50	1.71	0
16	-0.80	-0.50	1.71	0
17	-0.50	-0.50	1.71	0
18	-0.20	-0.50	1.71	0
19	0.10	-0.50	1.71	0
20	0.40	-0.50	1.71	0
21	0.70	-0.50	1.71	0
22	1.00	-0.50	1.71	0
23	-2.00	-0.50	1.43	0
24	-1.70	-0.50	1.43	0
25	-1.40	-0.50	1.43	0

- de visualiser les numéros des nœuds et des éléments et de faire une capture de la structure sous format jpg.

Options

Afficher Nœuds Azimut: < 60 >

Afficher Elements Elevation: < 30 >

OK Capture Image

- de faire des **zooms in** et **zooms out** et déplacement panoramique Pan ainsi que des rotations dans les trois directions.



Après avoir introduit les paramètres du matériau qui sont préalablement paramétrables, et les charges extérieures un fichier de données est généré automatiquement sous format 'fichier.txt' ou 'fichier.dat' ou encore 'fichier.gom' ;

Caractéristiques Matériaux

Matériau: Beton_E25

Nombre de Matériaux: 1

Module de Young (kN/m²): 25000000

Poids Volumique (kN/m3): 24.5166

Coef. Poisson: 0.2

Acier
Beton_E25
Beton_E20
Bois
Sol_Argile
Sol_Sable
Aluminium

Ainsi que deux fichiers distincts comportant les coordonnées ainsi que les connectivités 'coordXYZ.dat' et 'connect.dat' sont donnés.

Chargements

N°	Fx	Fy	Fz
1	0	0	0

Ajouter Charge

Effacer Charge

Interface générateur de maillage Hexaèdre 20 Nœuds.

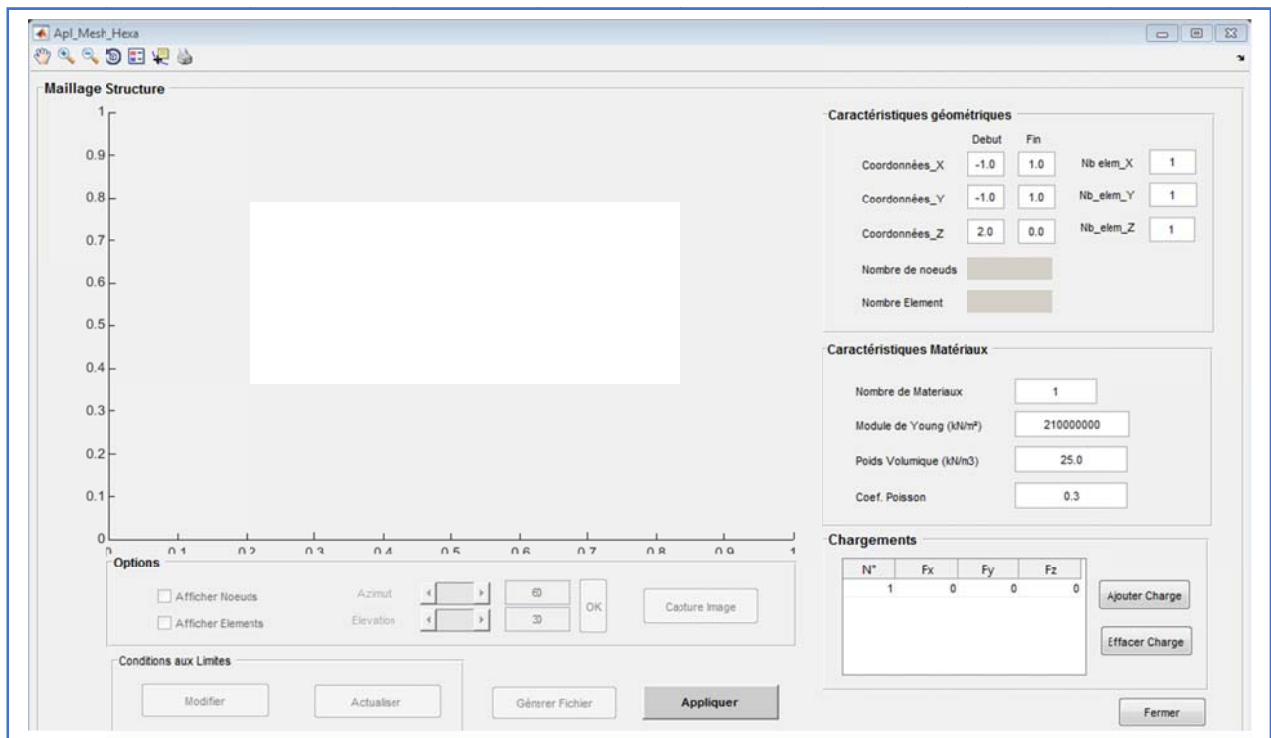


Figure 27 : Interface générateur de maillage Hexaèdre 20 Nœuds

Idem que pour l'élément hexaèdre à 8 nœuds après introduction des dimensions géométrique ainsi que le nombre de division dans les sens x y et z la construction du maillage est générée et la structure est affichée dans l'interface graphique.

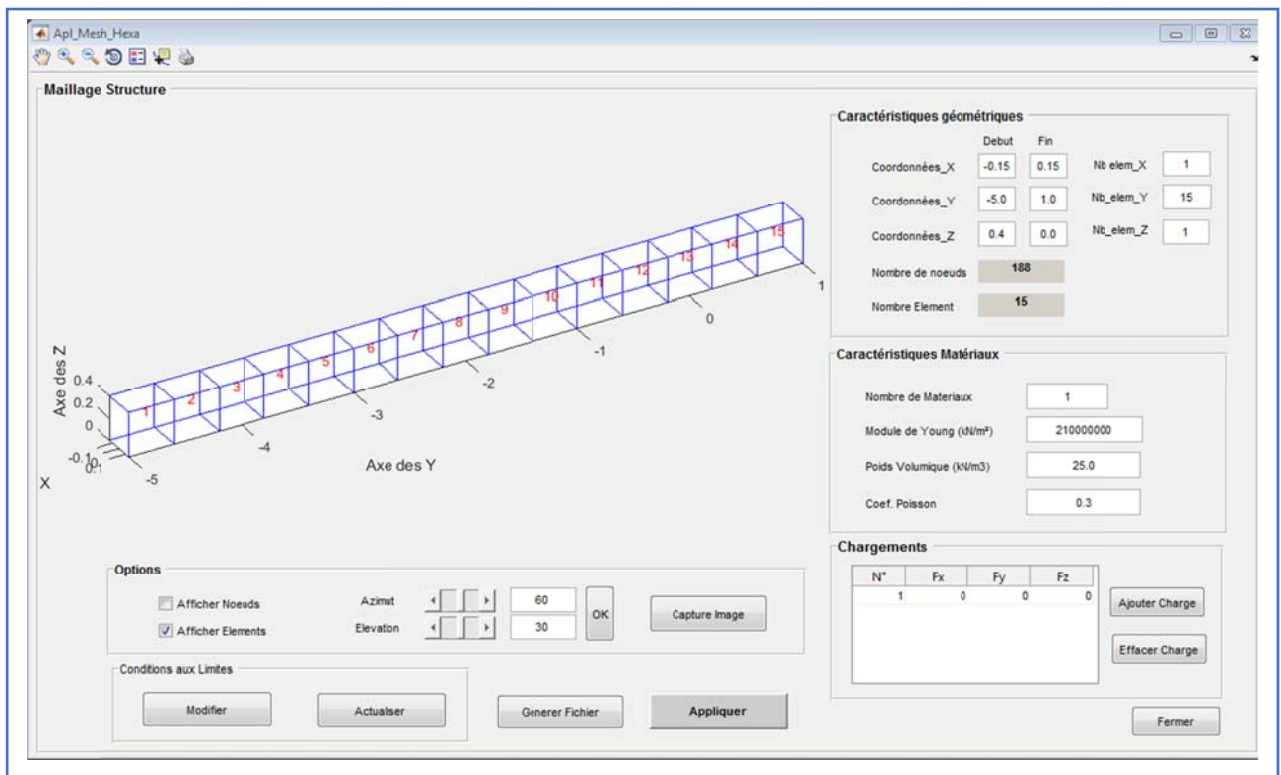


Figure 28 : Affichage du maillage de la structure

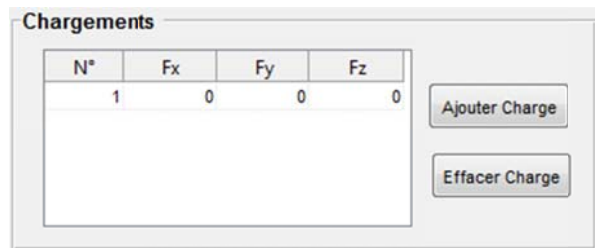
Possibilité de visualiser les numéros des nœuds et des éléments et de faire une capture de la structure sous format jpg.



- de faire des zooms in et out et un déplacement panoramique Pan ainsi que des rotations dans les trois directions.



Les charges extérieures peuvent être introduites en ajoutant une ligne dans le tableau des charges.



Paramètres du matériau de la structure

Un fichier de données est généré automatiquement sous format fichier.txt ou fichier.dat ou encore fichier.gom ainsi que deux fichiers distincts comportant les coordonnées ainsi que les connectivité coordXYZ.dat et connect.dat.

3.6 Apl_Élément barre

3.6.1 Interface du programme

L'interface du programme a l'ouverture est comme suit :

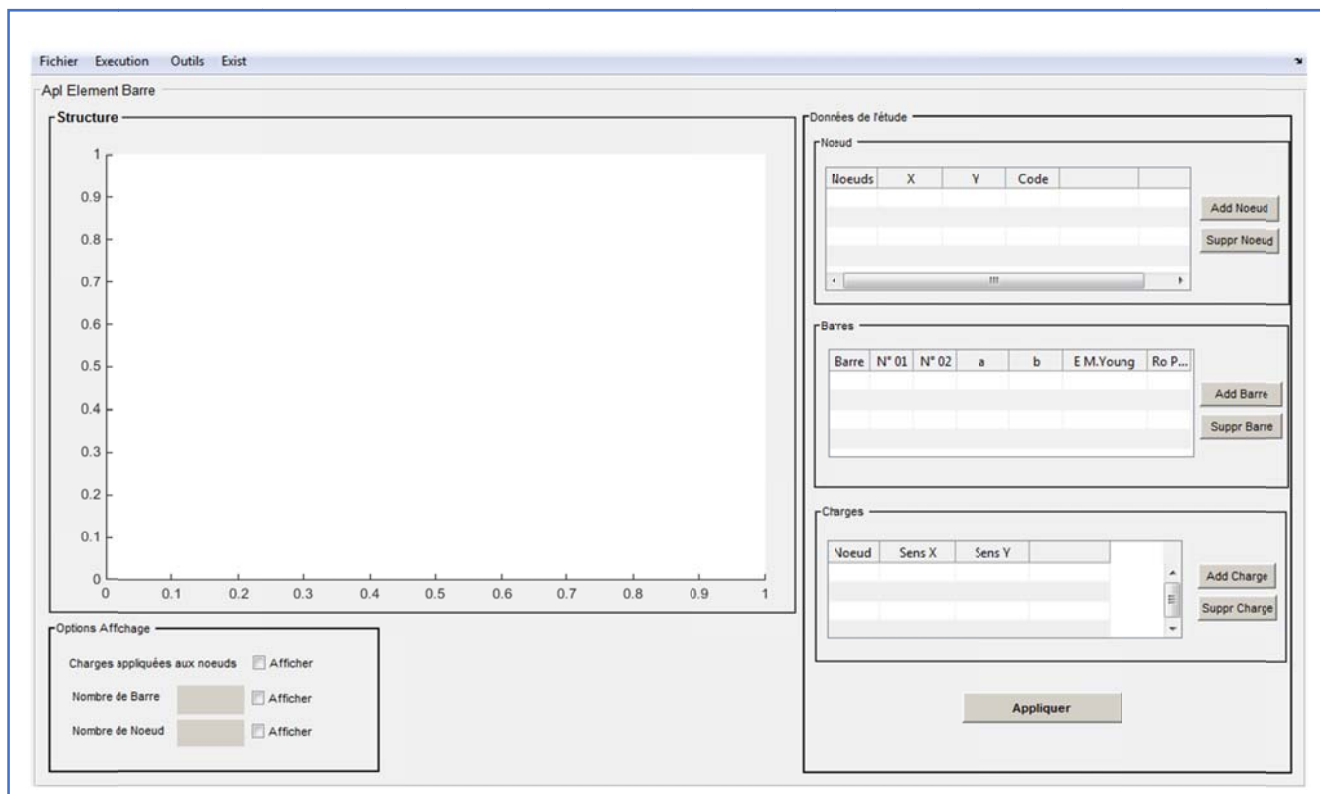
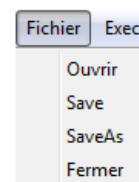


Figure 29 : L'interface du programme Apl_Design Barre

Le chargement du fichier de données de la structure préalablement défini et qui se présente sous l'extension .txt ou .dat se fait par la commande Ouvrir dans le menu fichier.



Après lecture des données ; la structure est affichée dans l'interface graphique et les données coordonnées, connectivités et chargements sont introduits dans les différents tableaux de l'interface.

Il sera possible d'afficher les numéros des nœuds ainsi que les numéros des éléments de même pour les vecteurs de charges.

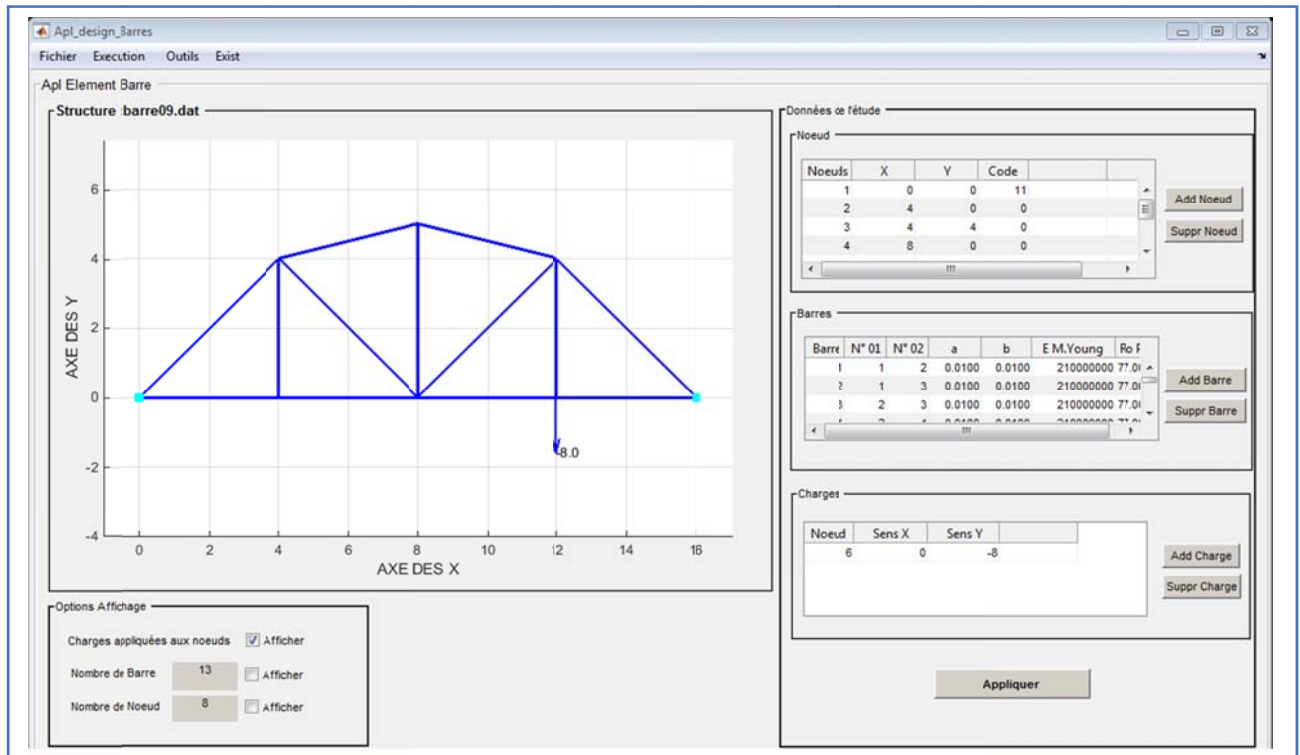


Figure 30 : Affichage de la structure

Après lancement des calculs les différents résultats déplacement, réactions d'appuis, ainsi que les efforts normaux sont affichés dans les tableaux et la déformée réelle de la structure est visualisée.

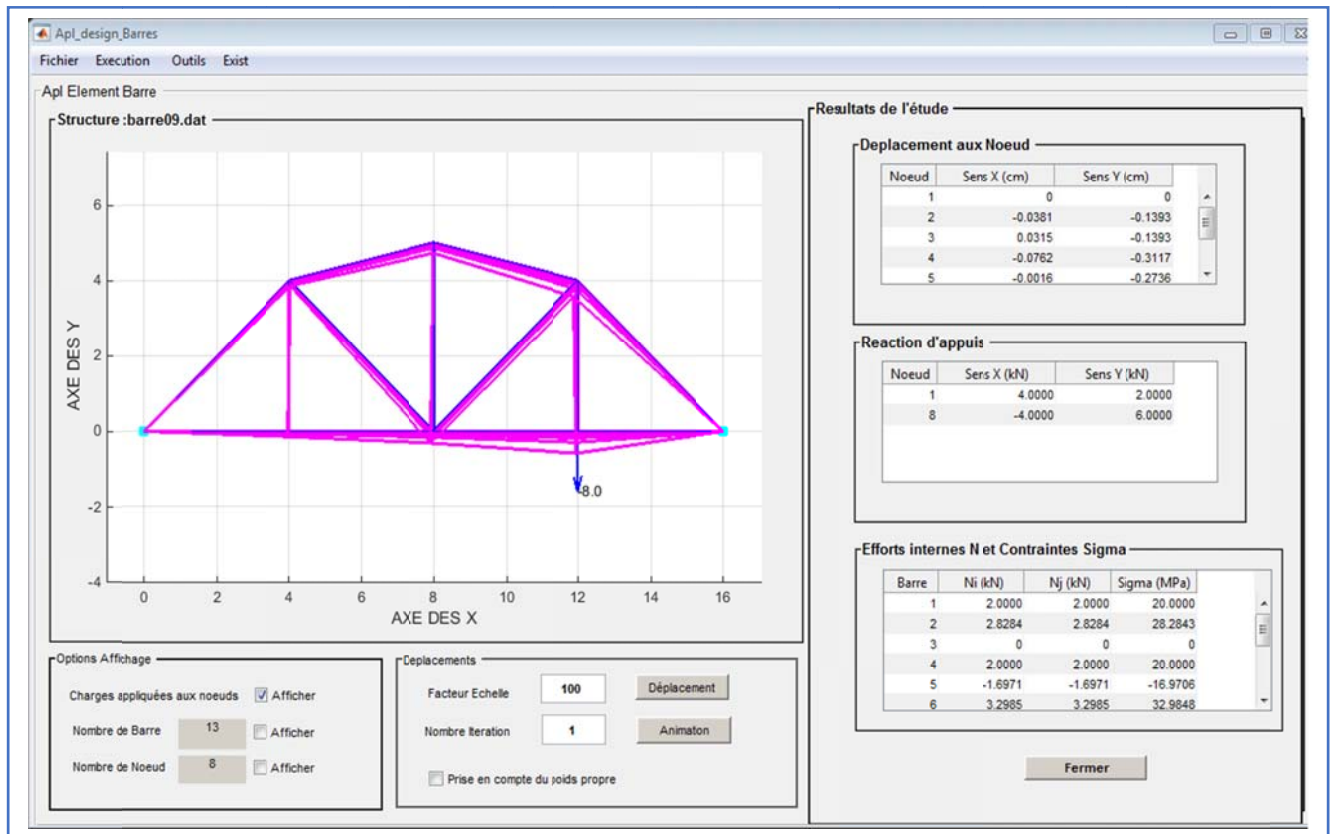


Figure 31 : Affichage des déplacements et des résultats

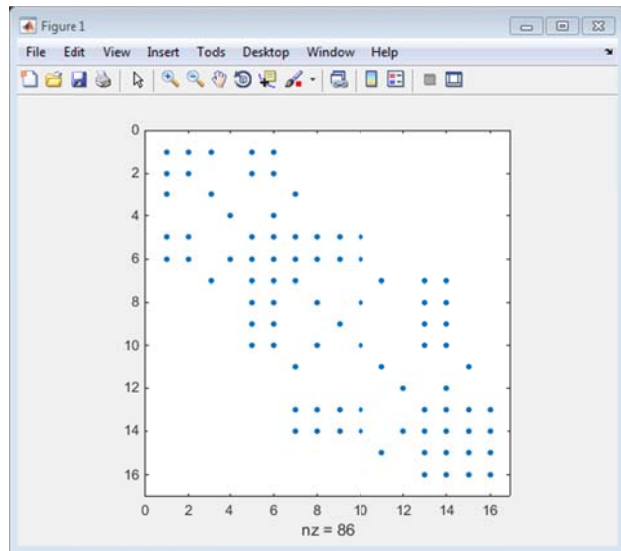
et il sera possible d'introduire un facteur d'échelle pour mieux visualiser les déplacements aux nœuds



Le bouton animation permet de voir les déplacements en introduisant le chargement de manier itérative et cela suivant un nombre d'itération

la matrice de rigidité global

Une note de calcul est générée
Avec Récapitulation des données



DONNEES DE LA STRUCTURE							
Nombre de Nœuds		8					
NOUEDES	X	Y	COD				
1	0.00000	0.00000	11				
2	4.00000	0.00000	0				
3	4.00000	4.00000	0				
4	8.00000	0.00000	0				
5	8.00000	5.00000	0				
6	12.00000	0.00000	0				
7	12.00000	4.00000	0				
8	16.00000	0.00000	11				
Nombre Elements		13					
elem	Ni	Nj	a (m)	b (m)	M.Young (MPa)	Poids Volumique (Kg/m2)	
1	1	2	0.01	0.01	210000000.0	77.00	
2	1	3	0.01	0.01	210000000.0	77.00	
3	2	3	0.01	0.01	210000000.0	77.00	
4	2	4	0.01	0.01	210000000.0	77.00	
5	4	3	0.01	0.01	210000000.0	77.00	
6	3	5	0.01	0.01	210000000.0	77.00	
7	4	5	0.01	0.01	210000000.0	77.00	
8	4	6	0.01	0.01	210000000.0	77.00	
9	4	7	0.01	0.01	210000000.0	77.00	
10	7	5	0.01	0.01	210000000.0	77.00	
11	6	7	0.01	0.01	210000000.0	77.00	
12	6	8	0.01	0.01	210000000.0	77.00	
13	8	7	0.01	0.01	210000000.0	77.00	

et les résultats du calculs

DEPLACEMENT AUX NOEUDS			
NOUEDES	UX (kN)	UY (kN)	
1	0.00000	0.00000	
2	-0.03810	-0.13930	
3	0.03155	-0.13930	
4	-0.07619	-0.31169	
5	-0.00163	-0.27359	
6	-0.03810	-0.58337	
7	-0.10774	-0.43099	
8	0.00000	0.00000	

REACTIONS AUX APPUIS		
NOUEDES	RX (kN)	FY (kN)
1	4.00000	2.00000
8	-4.00000	6.00000

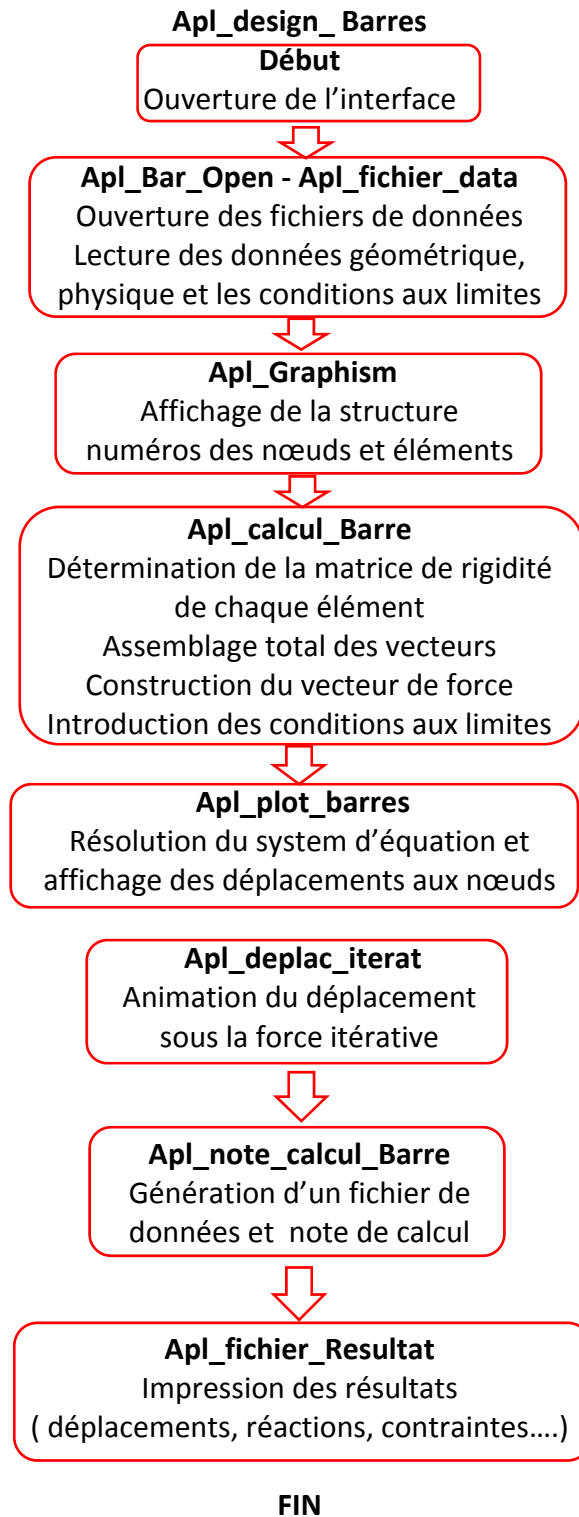
EFFOITS INTERNE DANS LES BARRES			
BARRE	Ni (kN)	Nj (kN)	SIGMA(MPa)
1	2.000	2.000	20.000
2	2.828	2.828	28.284
3	0.000	0.000	0.000
4	2.000	2.000	20.000
5	-1.697	-1.697	-16.971
6	3.298	3.298	32.985
7	-1.600	-1.600	-16.000
8	-2.000	-2.000	-20.000
9	3.960	3.960	39.598
10	3.298	3.298	32.985
11	-8.000	-8.000	-80.000
12	-2.000	-2.000	-20.000
13	8.485	8.485	84.853

3.6.2 Sous programmes

Sous programmes	Fonction
Apl_design_Barres.m	Interface du Programme principal
Apl_Bar_Open.m	Ouvrir le fichier de données
Apl_Bar_axes.m	Limitation des axes du graphe
Apl_plot_barres.m	Dessin de la structure
Apl_calcul_barre.m	Calcul des matrices de rigidités de masses et calcul les déplacements et efforts internes
Apl_deplac_iterat.m	Calcul du déplacements itératif
Apl_fichier_data.m	Construction du fichier de données dans le cas des modifications
Apl_fichier_Resultat.m	Construction du fichier des résultats
Apl_note_calcul_barre.m	Élaboration d'une note de calcul finale
affich_matrice.m	Affiche la matrice de rigidité
Apl_dynam_barre.m	Résolution de l'équation du mouvement par la méthode de Newmark
Apl_eig_perodbar.m	Extraction des fréquences et modes propres

Tableau 6 : sous programmes Apl_Design_Barre

3.6.3 Organigramme de résolution



3.7 Apl Élément Poutre

3.7.1 Interface du programme Apl_Poutre

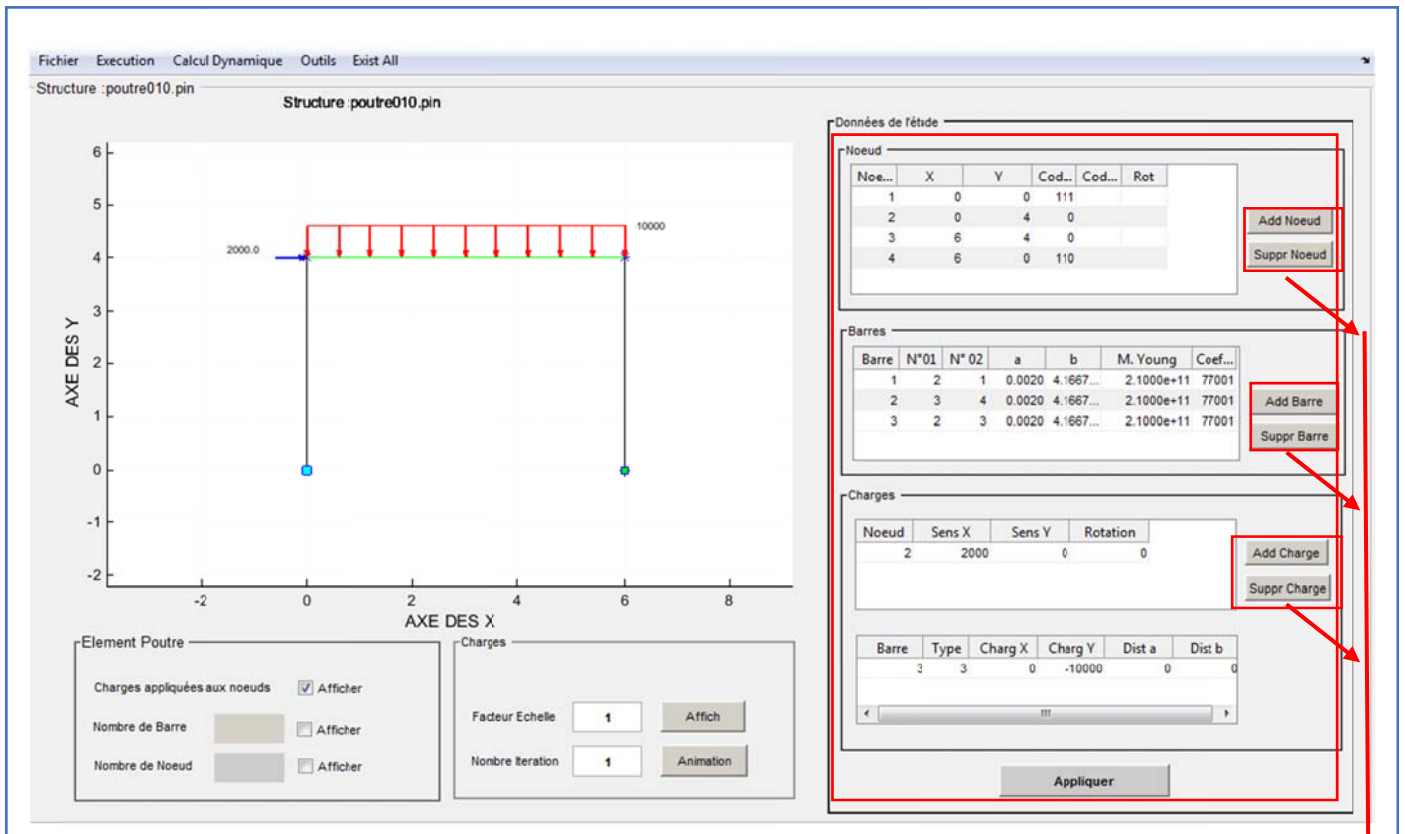


Figure 32 : interface Apl_design_Poutre

Il est possible de modifier les structures soit par ajout de nœuds ou d'éléments poutre ainsi que les charges et leurs positions (nœuds sur lequel sont appliqué

Partie Statique

Après lancement des calculs la déformée à l'échelle réelle de la structure est affichée dans l'interface graphique avec les résultats des déplacements en chaque nœuds et réactions d'appuis ainsi que les efforts internes.

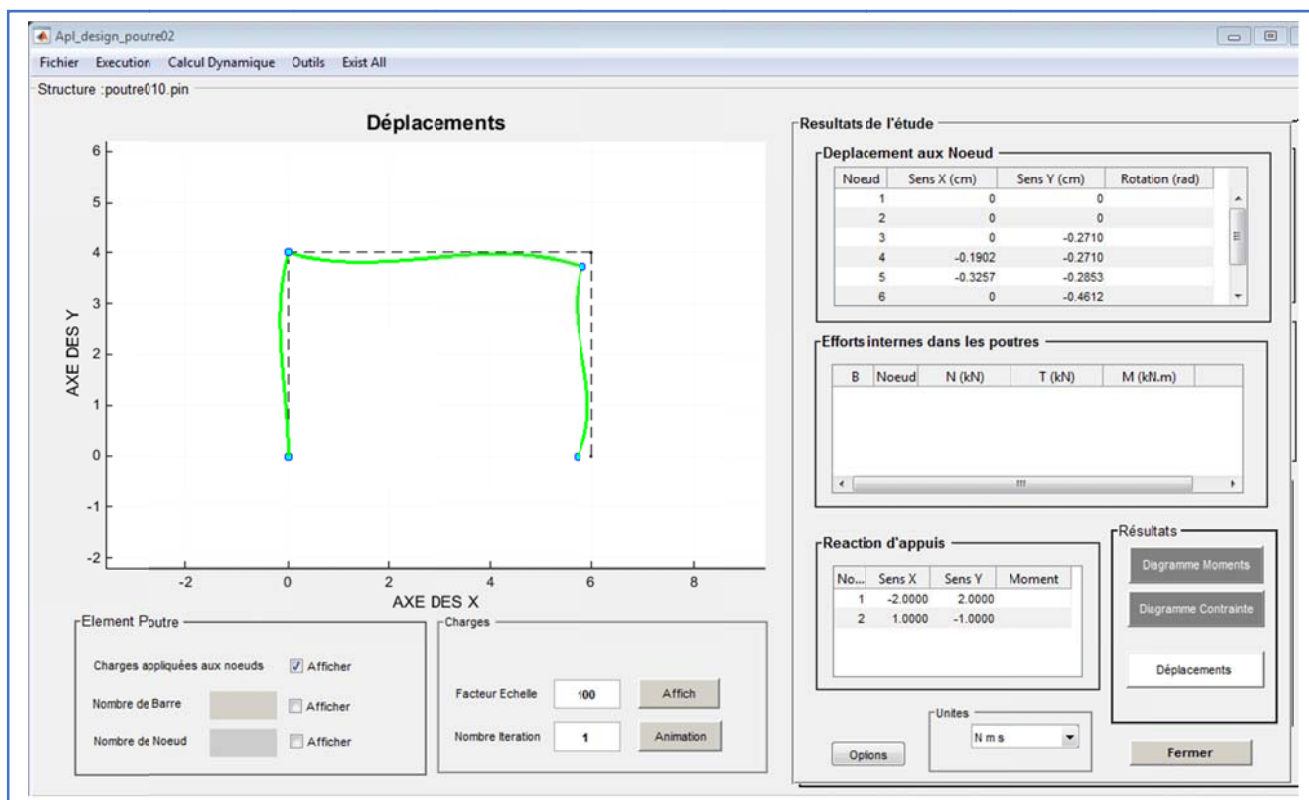


Figure 33 d : éformée de la structure

Un module a été ajouté pour visualiser les diagrammes des efforts internes ici Diagramme des efforts normaux avec les valeur de l'effort normal le long des poutres .

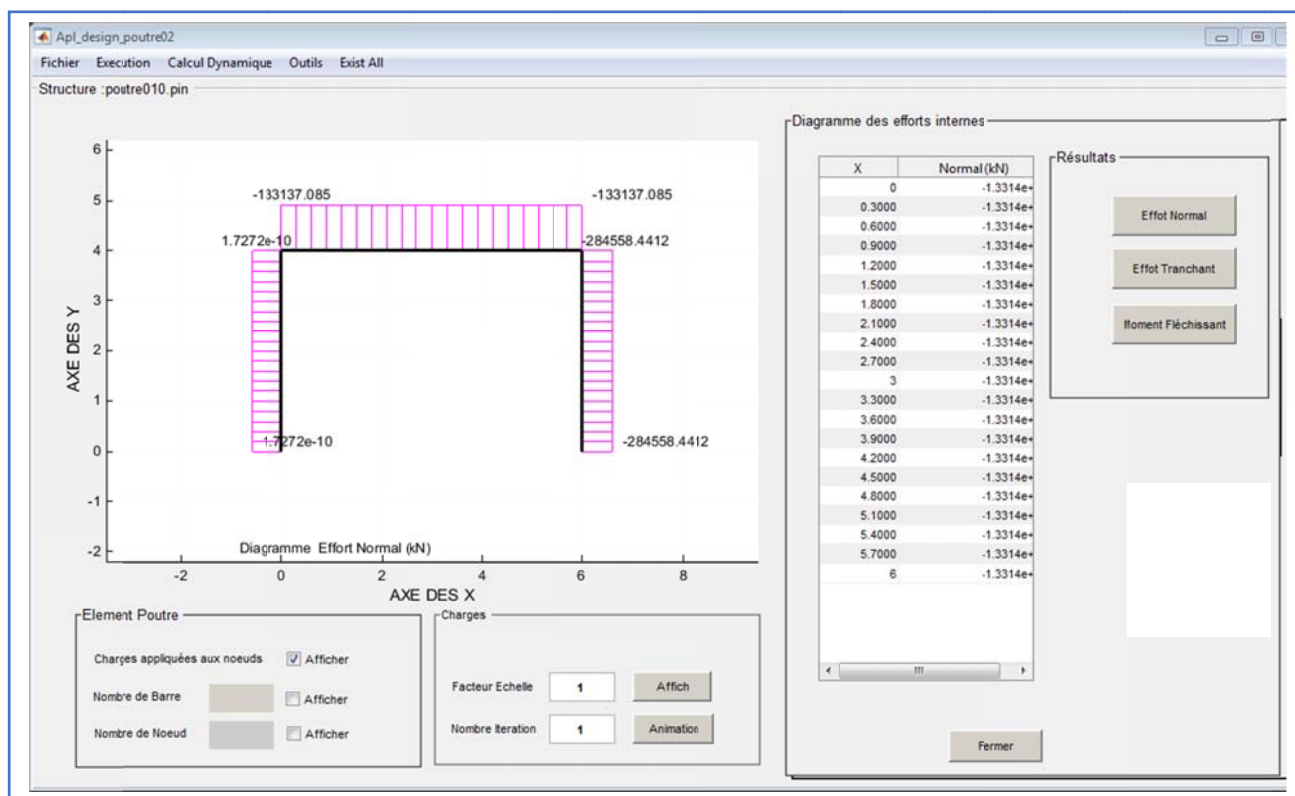


Figure 34 : Affichage Diagramme des efforts normaux

Diagramme du moments fléchissant le long des poutres

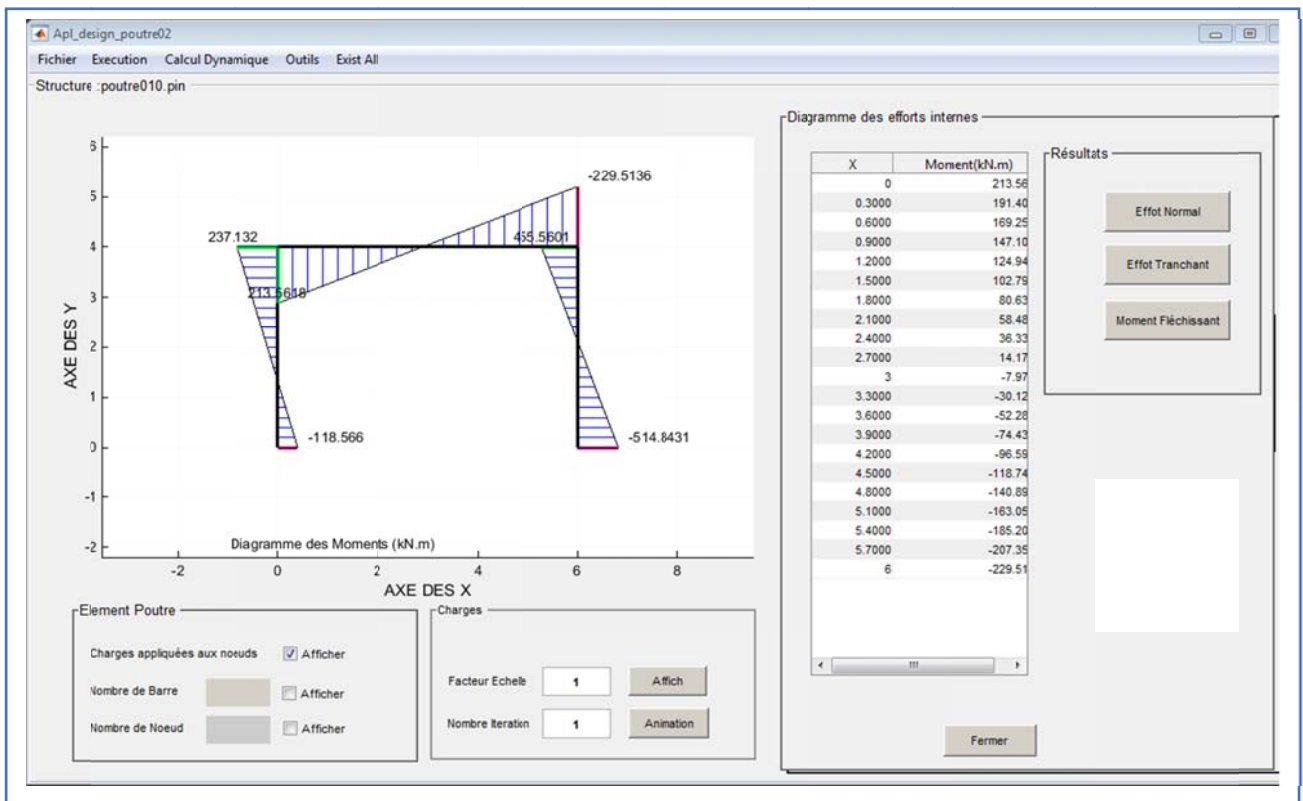


Figure 35 : Diagramme du moments fléchissant Affichage

Diagramme des efforts tranchants le long de la poutre

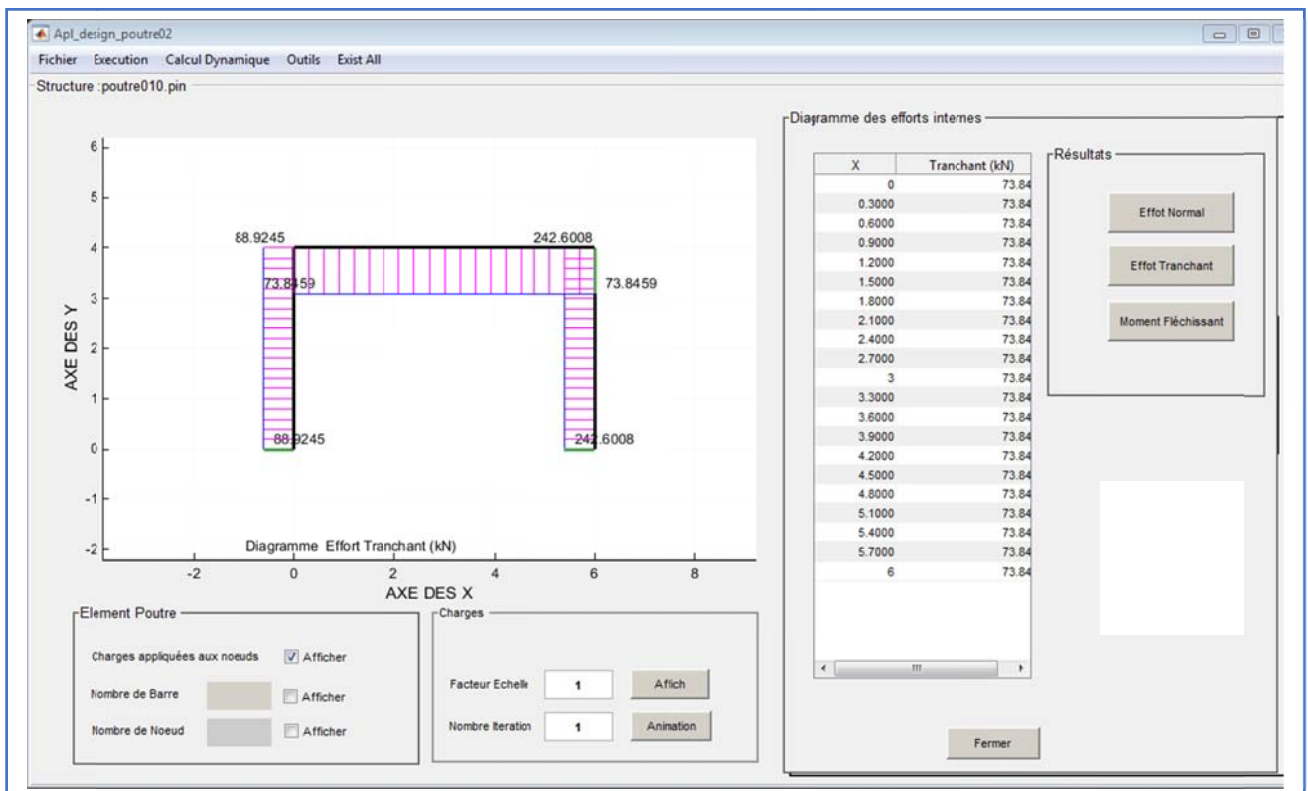


Figure 36 : Affichage Diagramme des efforts tranchants

Partie dynamique

Sous l'effet d'un séisme comme exemple séisme El Centro

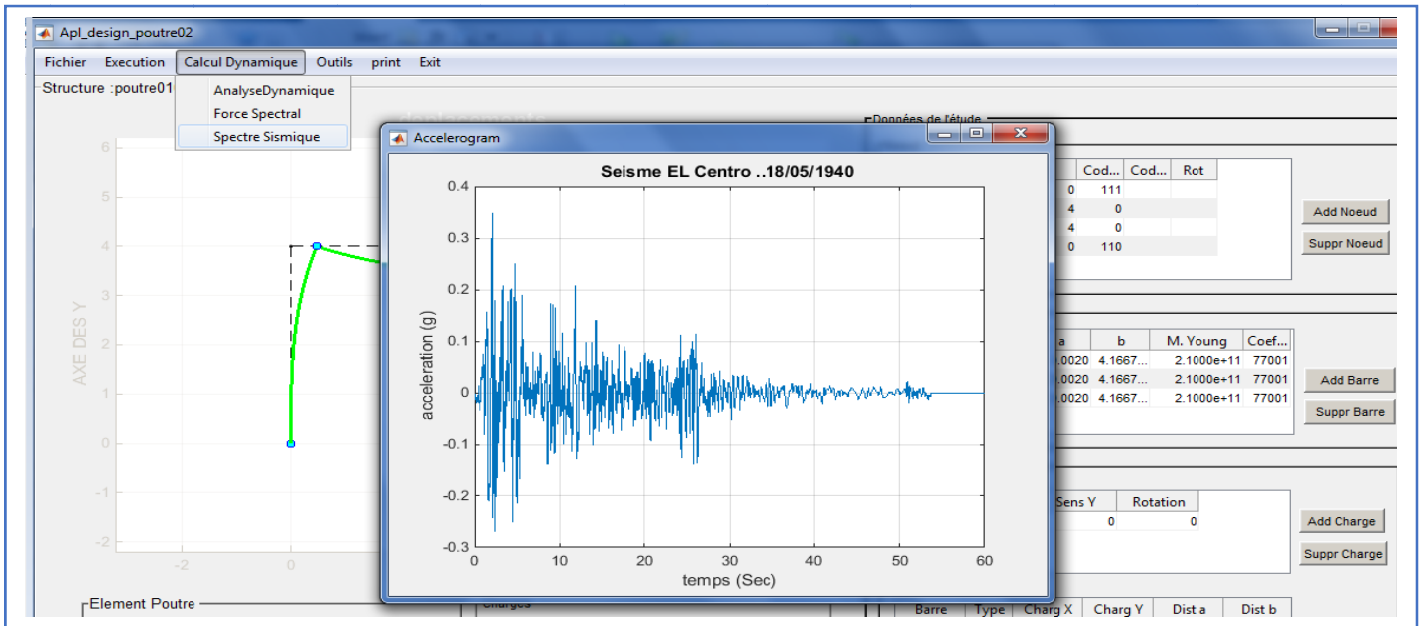


Figure 37 : Affichage du spectre de réponse

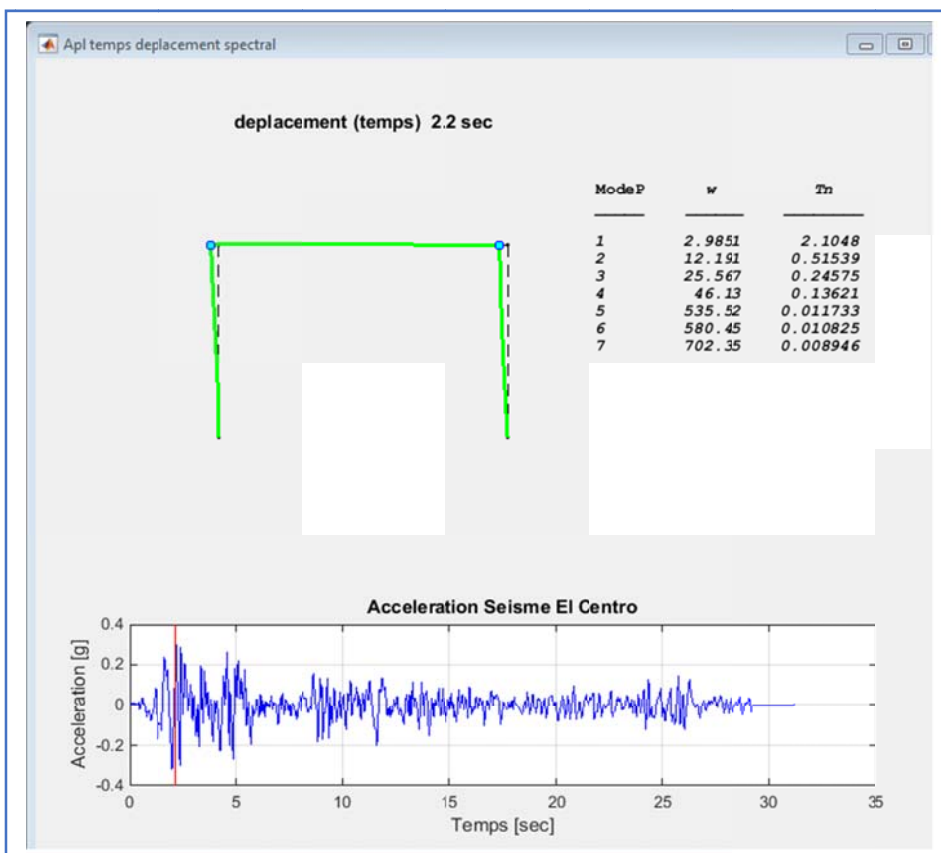
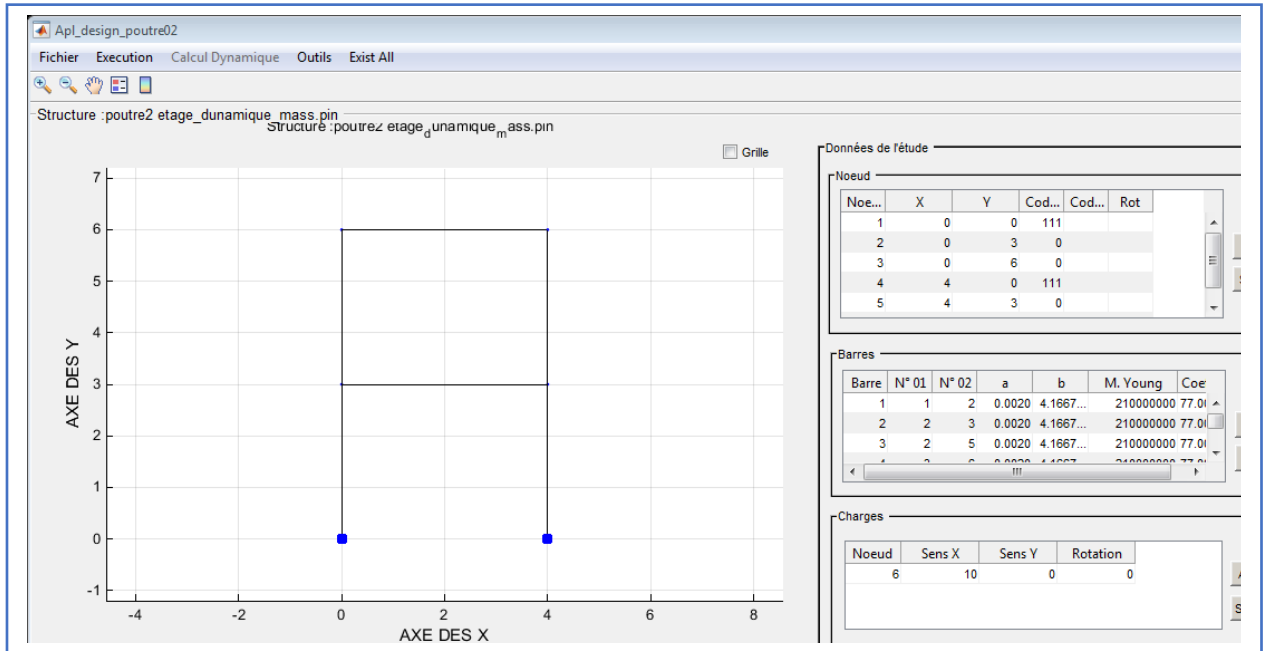


Figure 38 : Déformée animée de la structure durant un séisme

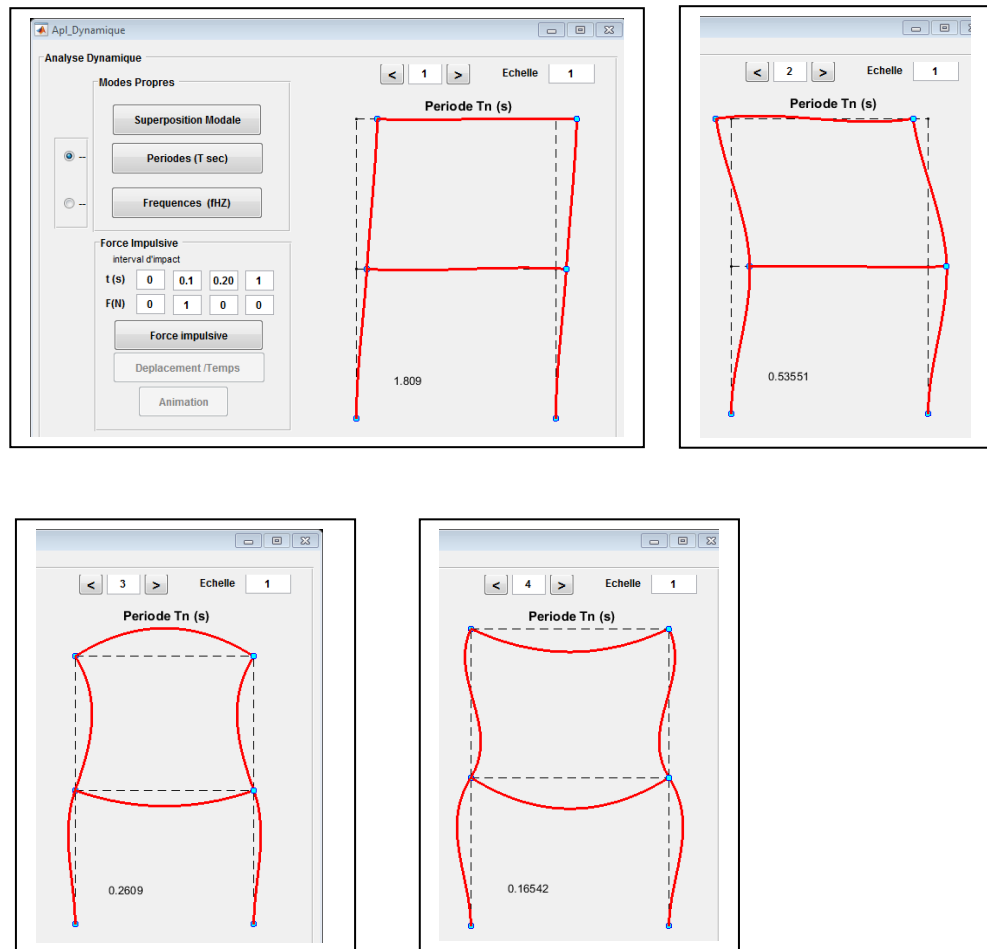
- Le dessin suivant est animé montrant le mouvement ou le comportement de la structure pendant le séisme
- La ligne rouge indique l'incrément de temps concerné par le dessin au-dessus.
- Les modes propres sont affichés à droite de la structure en mouvement

Partie dynamique

Soit un portique de 2 étages



Les modes propres de la structure sont données comme suit



Sous l'effet d'un séisme

On prend comme exemple le séisme d' El Centro où il est possible de charger les données spectrales à partir d'un fichier externe et de tracer le spectre de réponse

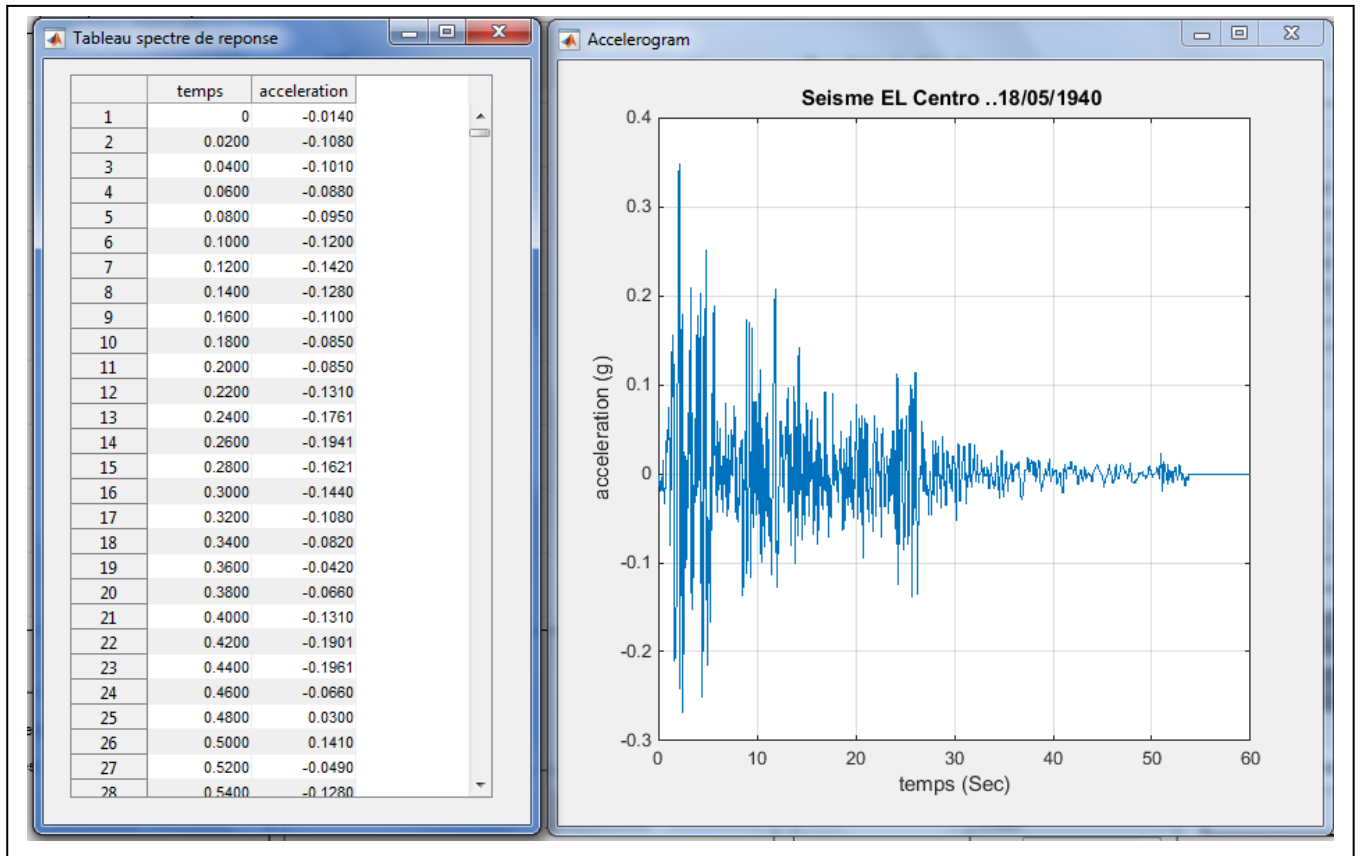
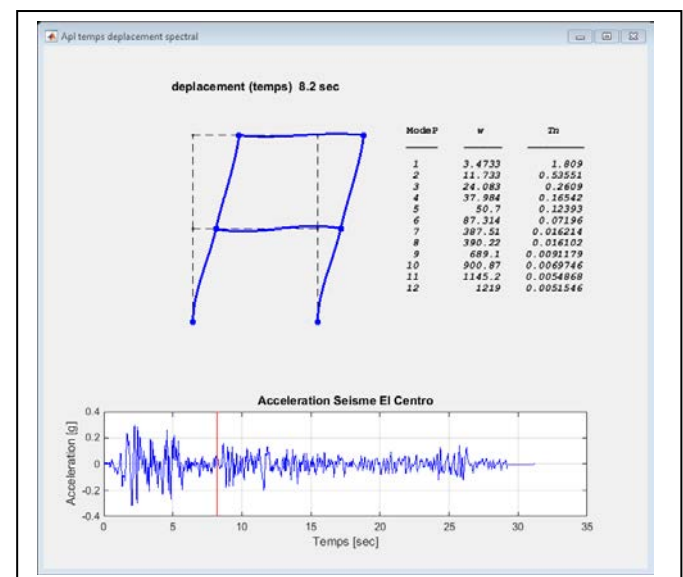
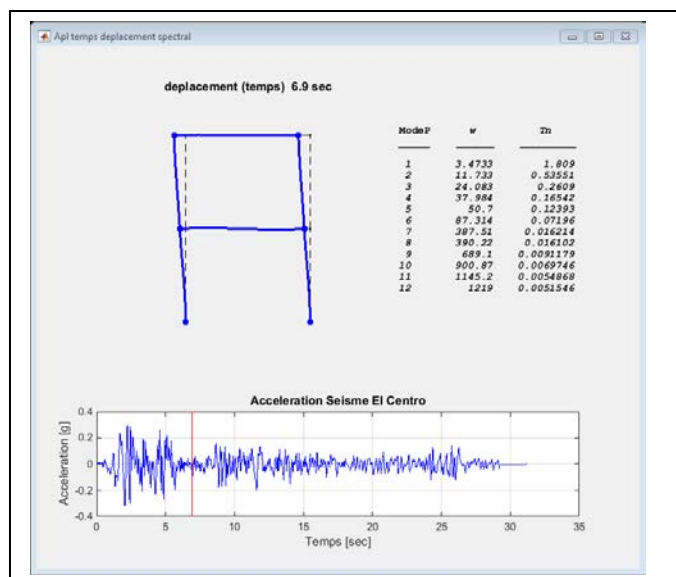
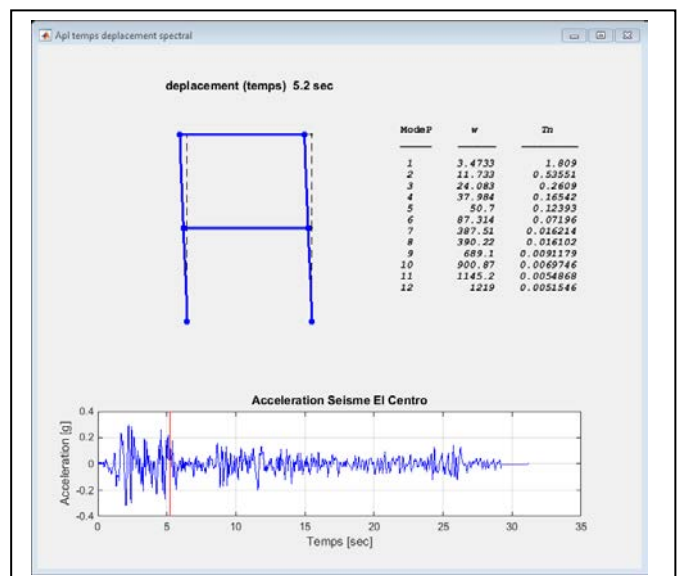
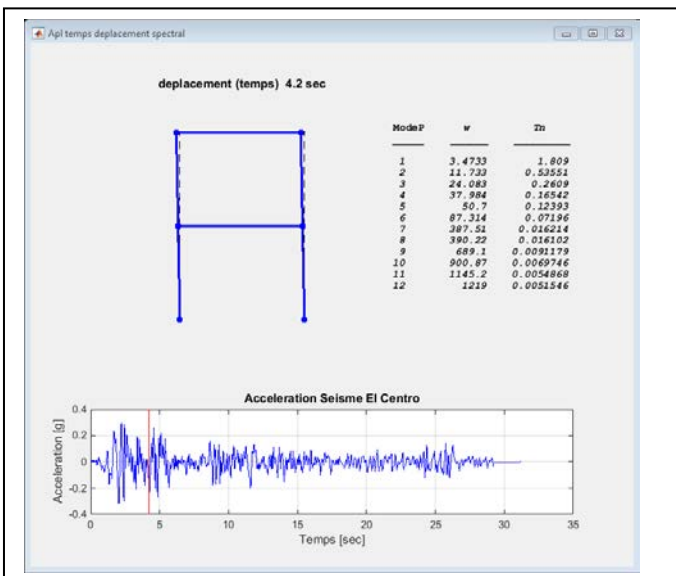
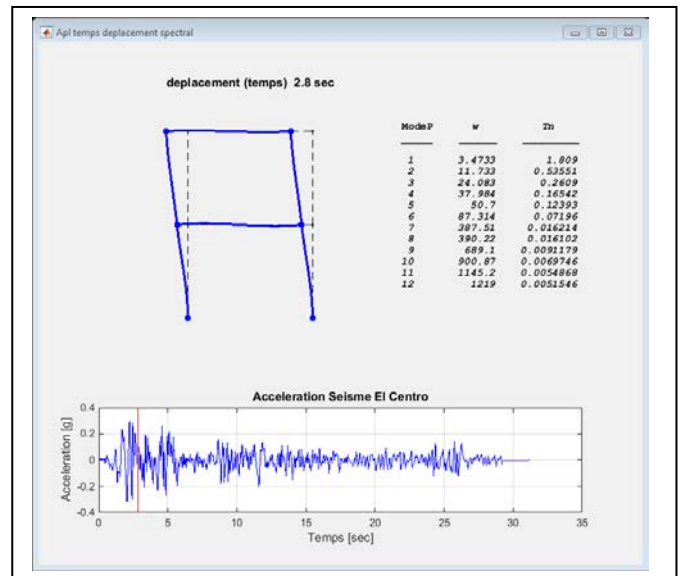
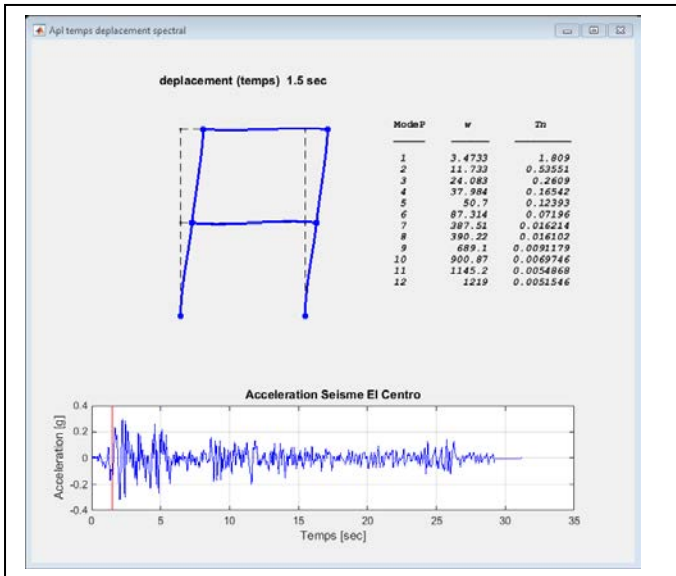
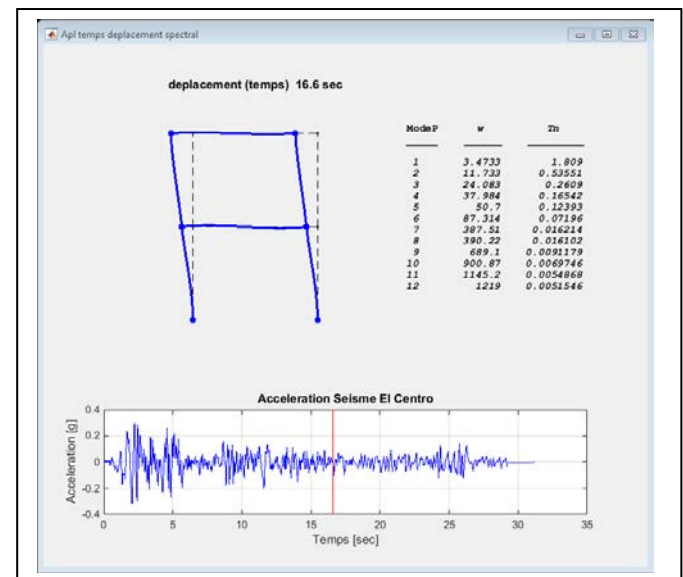
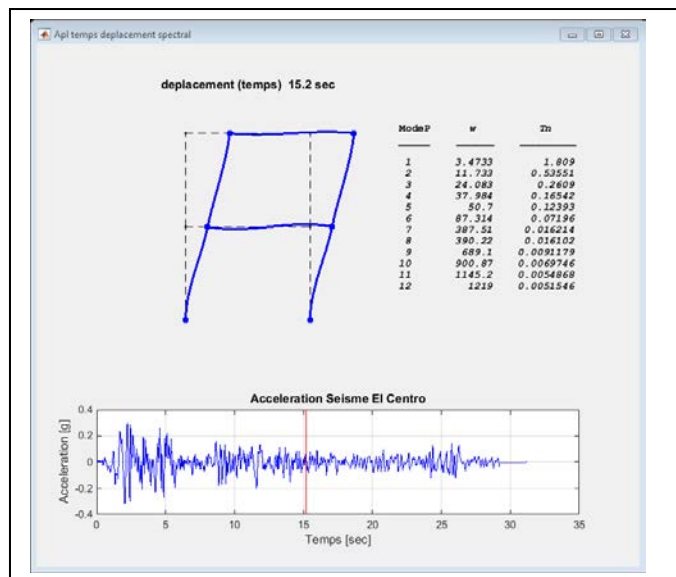
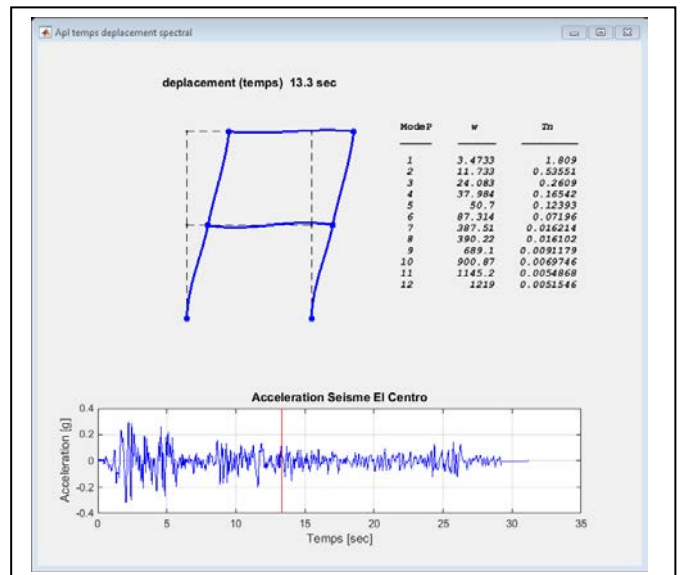
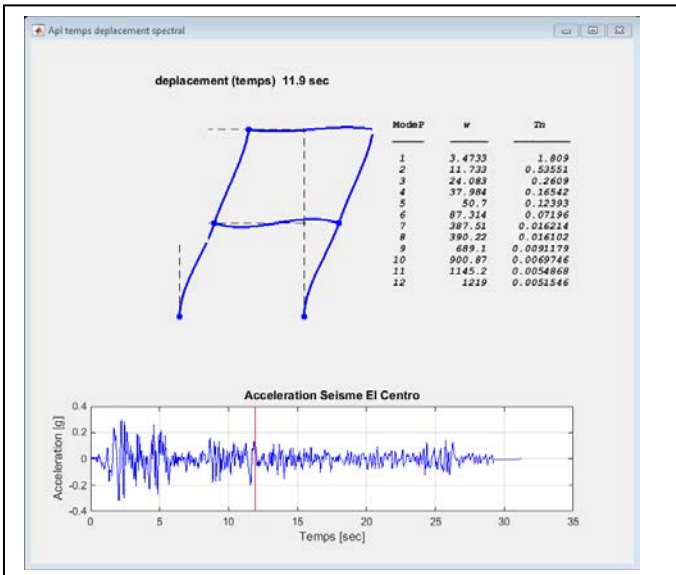
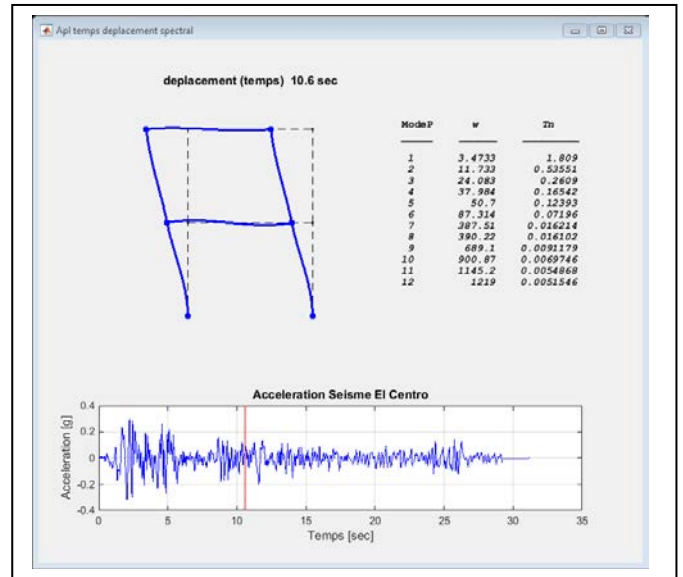
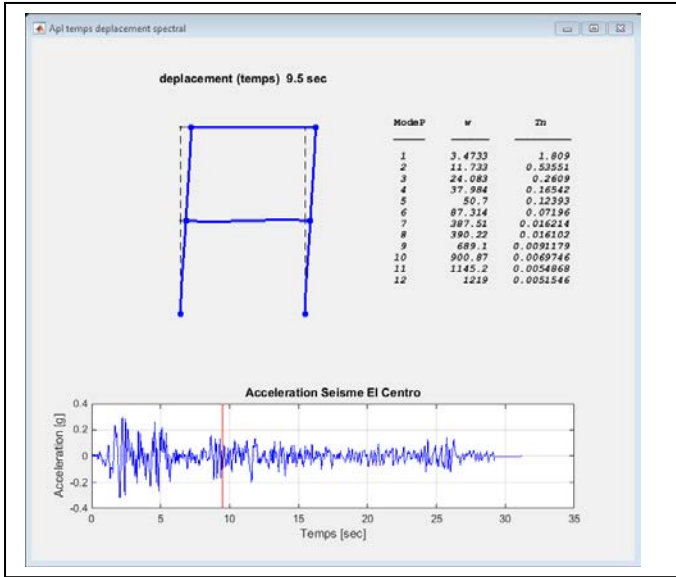


Figure 1 : Affichage du spectre de réponse

- Les figures suivantes montrent par animation le mouvement ou le comportement de la structure pendant le séisme.
- La ligne rouge indique l'incrément de temps concerné par le dessin au-dessus.
- Les modes propres sont affichés à droite de la structure en mouvement

Déformation du portique en fonction du temps sous l'effet du séisme est données comme suit





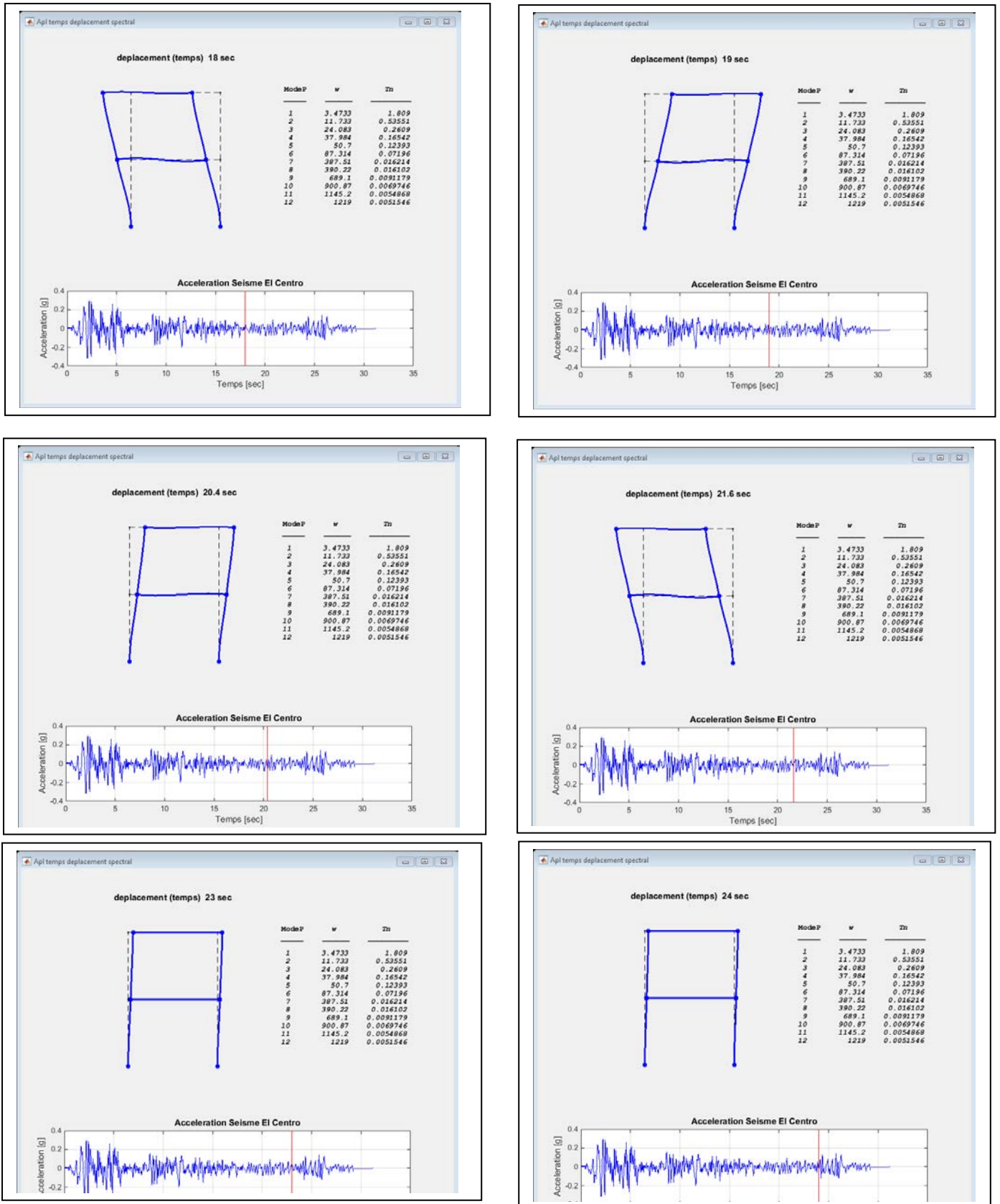


Figure 2 : Déformée animée de la structure durant un séisme

Force impulsive

Soit la structure suivante

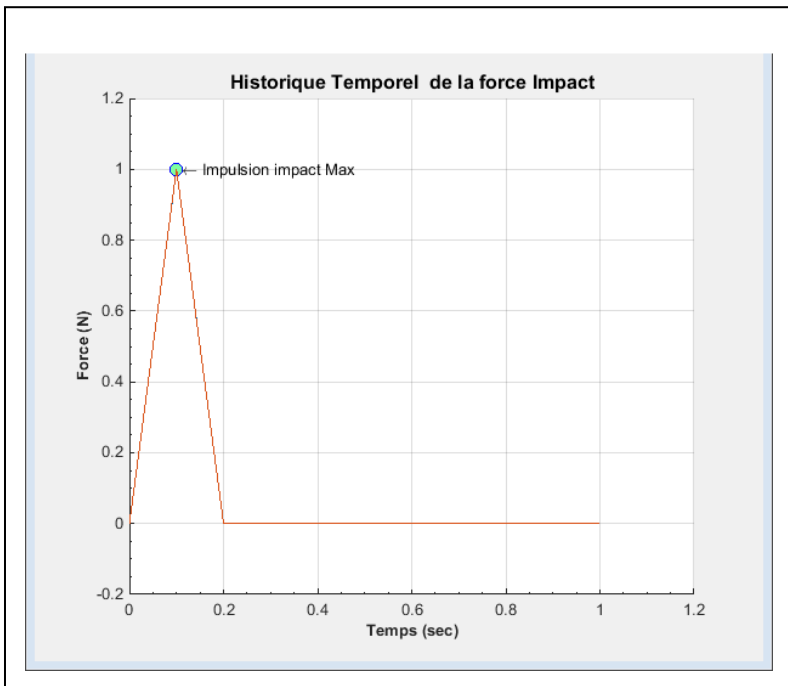
The screenshot shows the Apl Design MARS software interface. The main window displays a 2D structure with nodes 1 through 5 and elements 1 through 4. A force of 1000.0 is applied at node 3. The interface includes a menu bar (Fichier, Execution, Calcul Dynamique, Outils, Exist All), a toolbar, and a grid. On the right, there are three data tables: 'Noeud', 'Barres', and 'Charges'. Below these tables are control buttons for 'Add', 'Suppr', and 'Afficher' for each category, along with a 'Structure' button and an 'Animation' button. At the bottom right, there is an 'Appliquer' button.

Noeud					
Noe...	X	Y	Cod...	Cod...	Rot
1	0	0	111		
2	0.5000	1.5000	0		
3	1	3	0		
4	1.5000	3	0		
5	2	3	10		

Barre	N° 01	N° 02	a	b	M. Young	Coef...
1	1	2	0.0010	1.7100...	3.0000e+10	2500
2	2	3	0.0010	1.7100...	3.0000e+10	2500
3	3	4	7.6400...	8.0000...	3.0000e+10	2500
4	4	5	7.6400...	8.0000...	3.0000e+10	2500

Noeud	Sens X	Sens Y	Rotation
3	1000	0	0

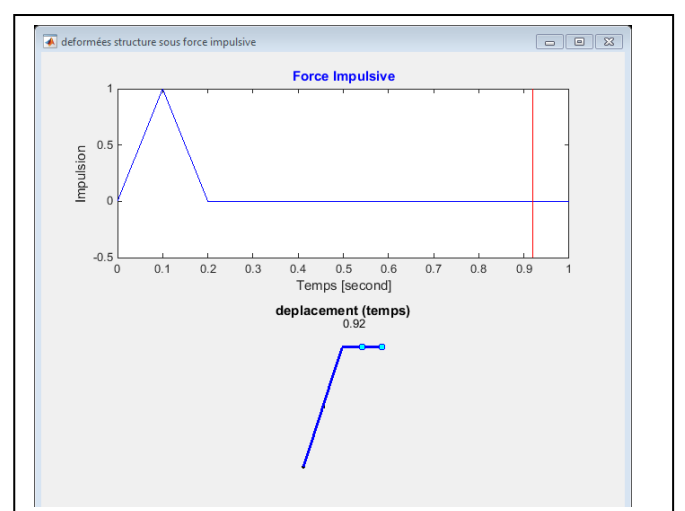
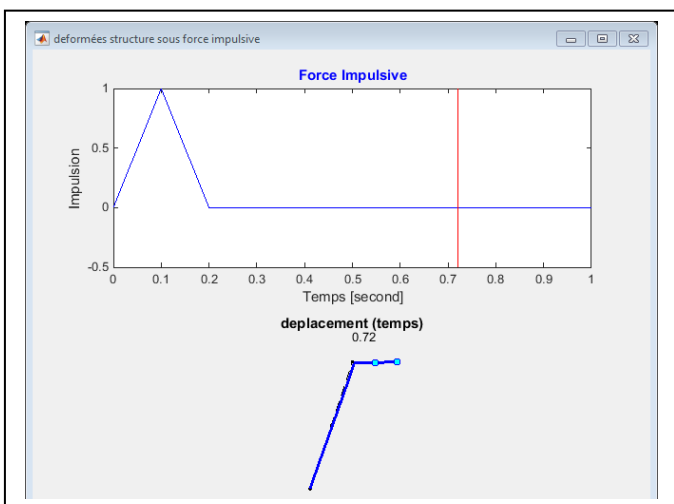
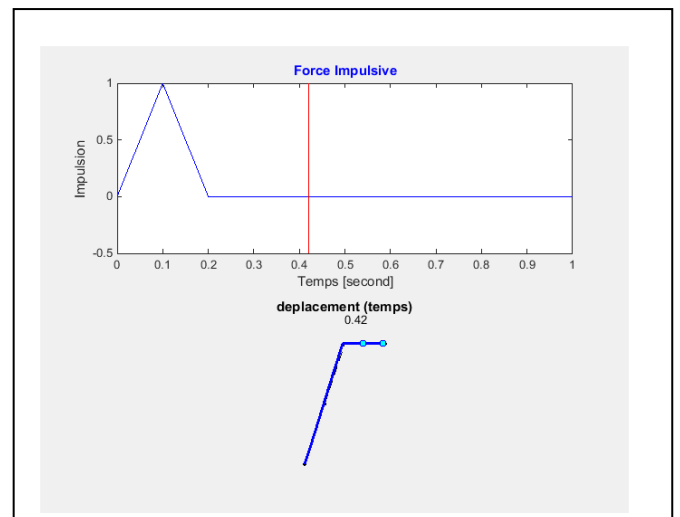
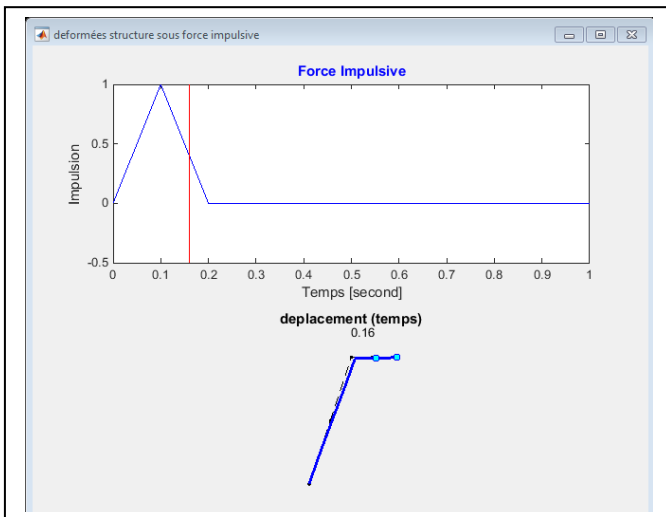
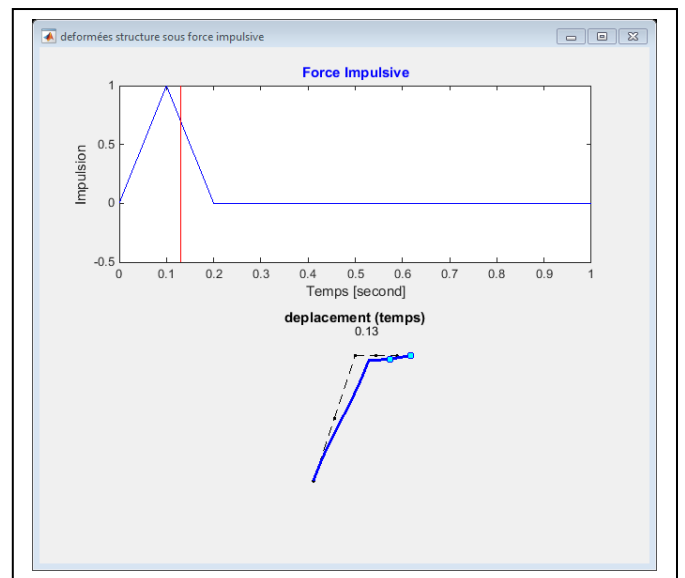
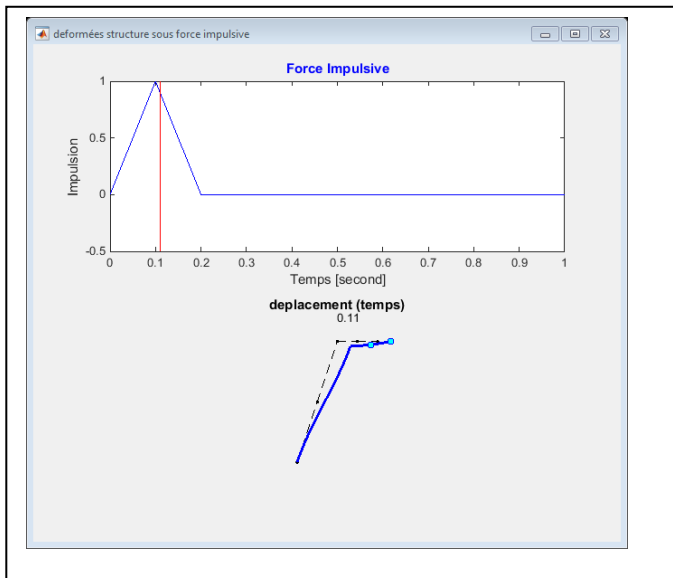
Sous une force impulsive (explosion par exemple) qu'il est possible de paramétrer et de voir le graphe en fonction du temps comme suit



The control panel for 'Force Impulsive' includes the following elements:

- Force Impulsive** (Section Header)
- interval d'impact** (Label)
- t (s)** (Time input): 0, 0.1, 0.20, 1
- F (N)** (Force input): 0, 1, 0, 0
- Force impulsive** (Button)
- Deplacement / Temps** (Button)
- Animation** (Button)

La déformée animée sous cette force impulsive est donnée comme suit



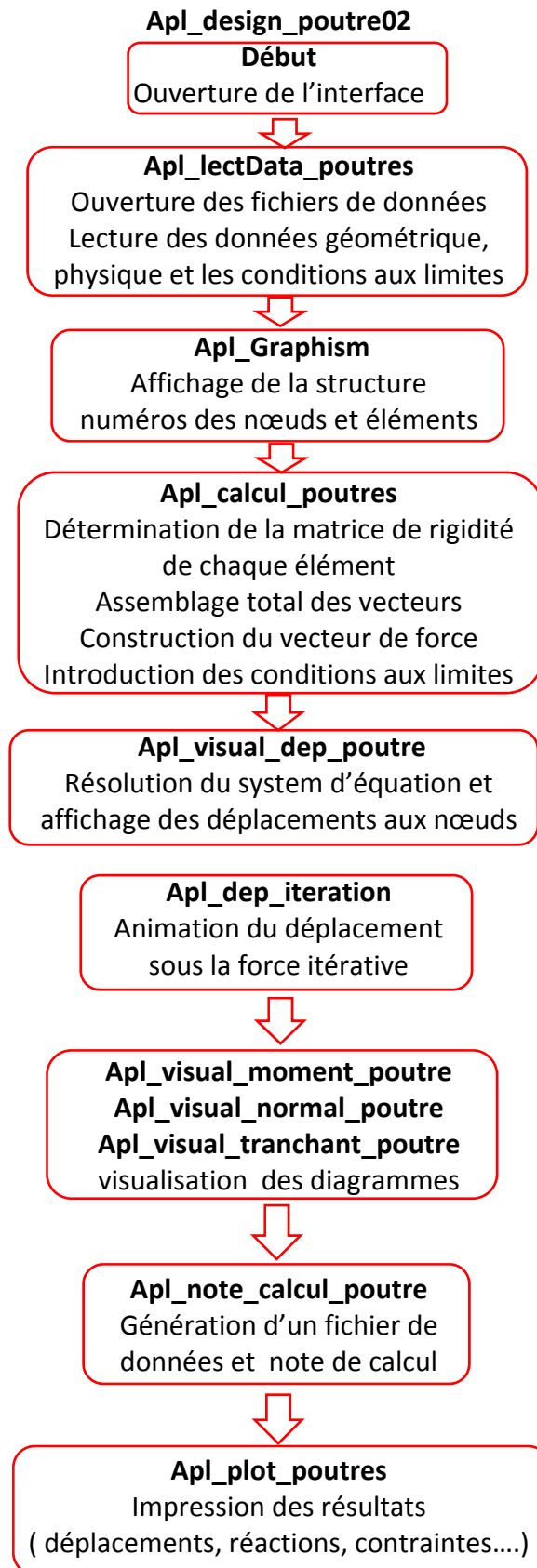
Déformée animée de la structure durant une force impulsive

3.7.2 Sous programmes

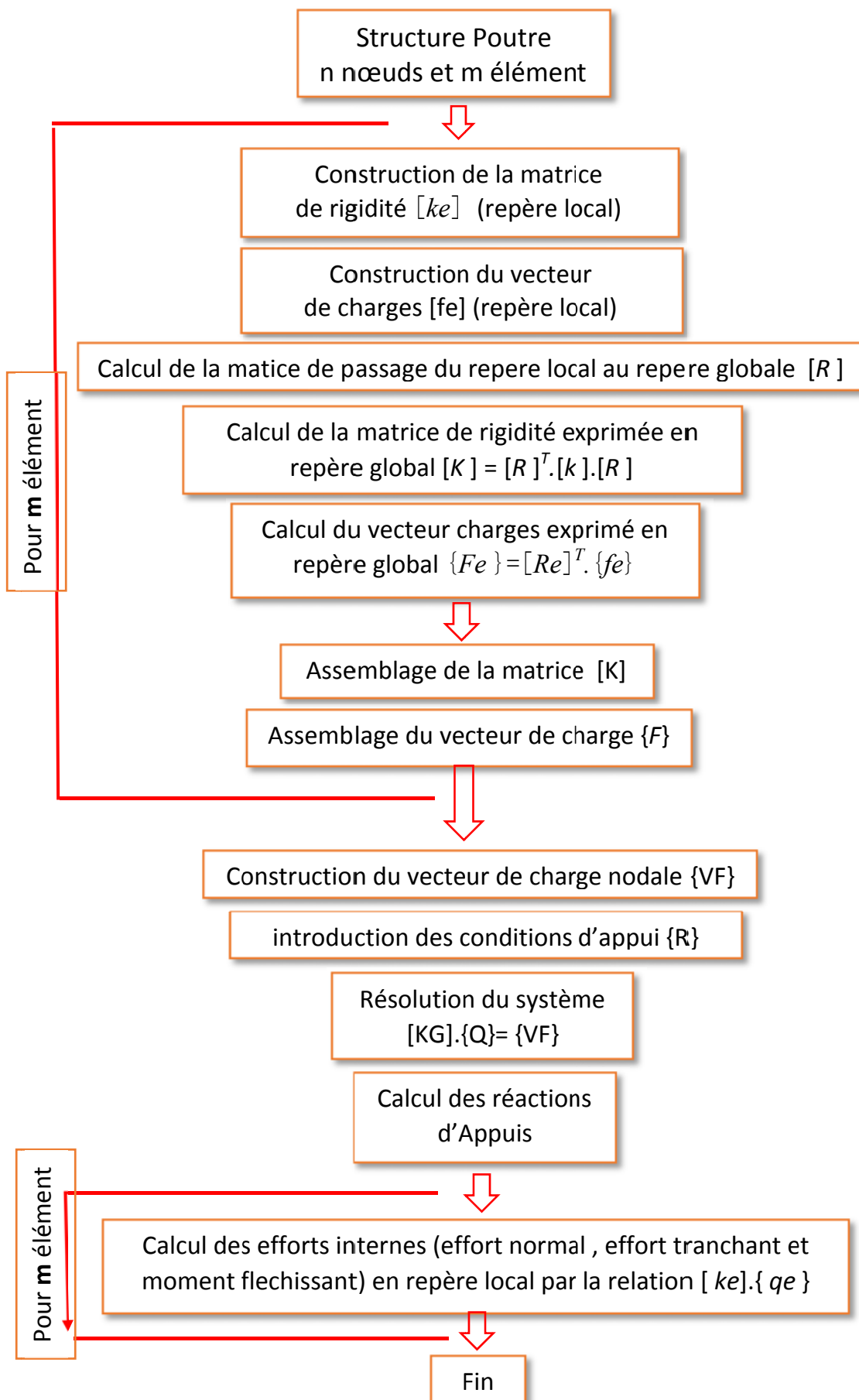
Sous programmes	Fonction
Apl_axes.m	Limitation des axes du graphe
Apl_calcul_poutres.m	Programme principal calcul des matrices de rigidités de masses et calcul les déplacements et efforts internes
Apl_ch_repartie_poutre.m	Dessin des charges réparties
Apl_charge_poutre.m	Construction du vecteur de charge
Apl_dep_iteration.m	Calcul du déplacement itératif
Apl_design_poutre02.m	Programme principal
Apl_Graphism.m	Module graphisme
Apl_lectData_poutres.m	Lecture du fichier de données poutre
Apl_note_calcul_poutre.m	Elaboration de la note de calcul
Apl_plot_ch_node_poutre.m	Dessin des charges aux nœuds
Apl_plot_poutres.m	Dessin de la structure
Apl_visual_dep_poutre.m	Visualisation de la déformée
Apl_visual_moment_poutre.m	Visualisation du moment fléchissant
Apl_visual_normal_poutre.m	Visualisation de l'effort normal
Apl_visual_tranchant_poutre.m	Visualisation de l'effort tranchant

Tableau 7 : Subroutines Apl_design_Poutre

3.7.3 Organigramme de résolution et sousroutines



Organigramme général de résolution



3.8 Apl Élément triangulaire T6

3.8.1 Interface du programme

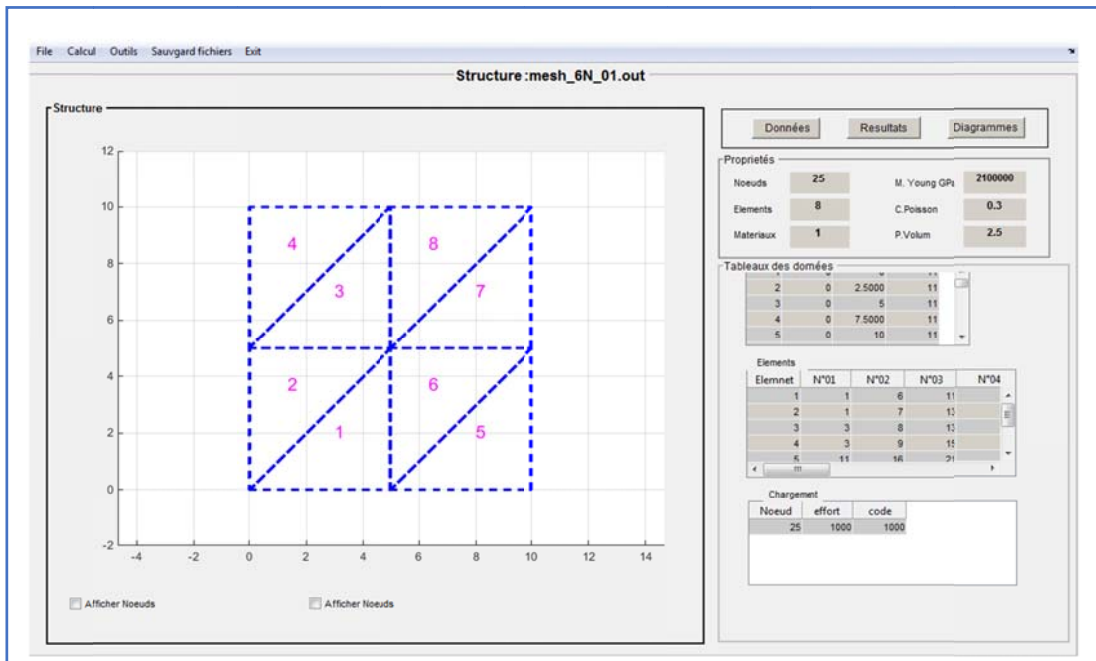


Figure 39 : Interface Apl_Design_triangle T6

Après lancement des calculs le déplacement à l'échelle réelle de la structure est affiché dans l'interface graphique

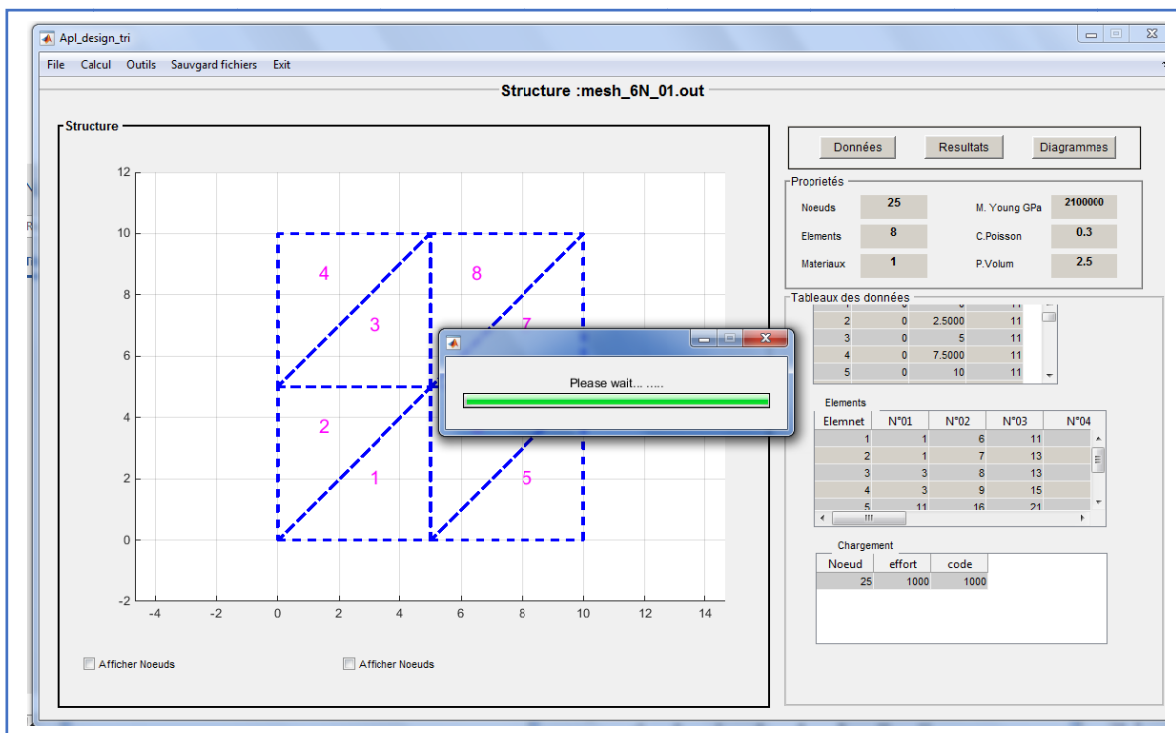


Figure 40 : lancement des calculs

Visualisation des déplacements et des contraintes

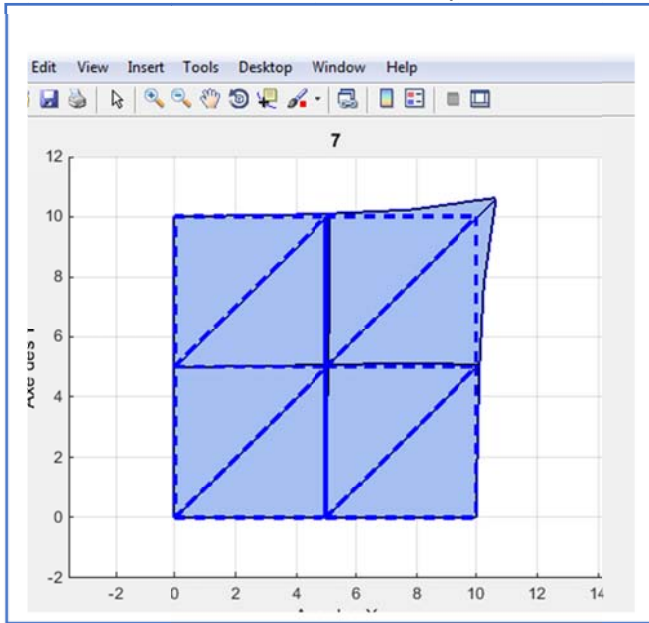


Figure 41 : animation des déplacements

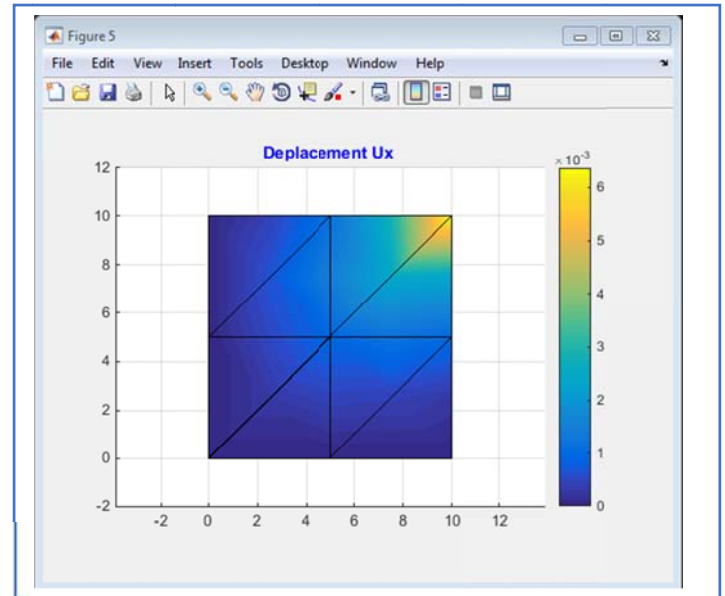


Figure 42 : déplacement U_x

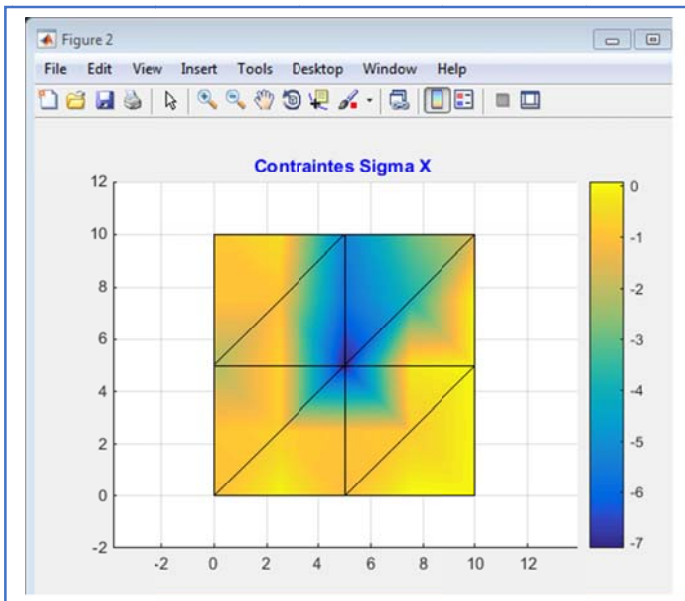


Figure 43 : Contrainte σ_x

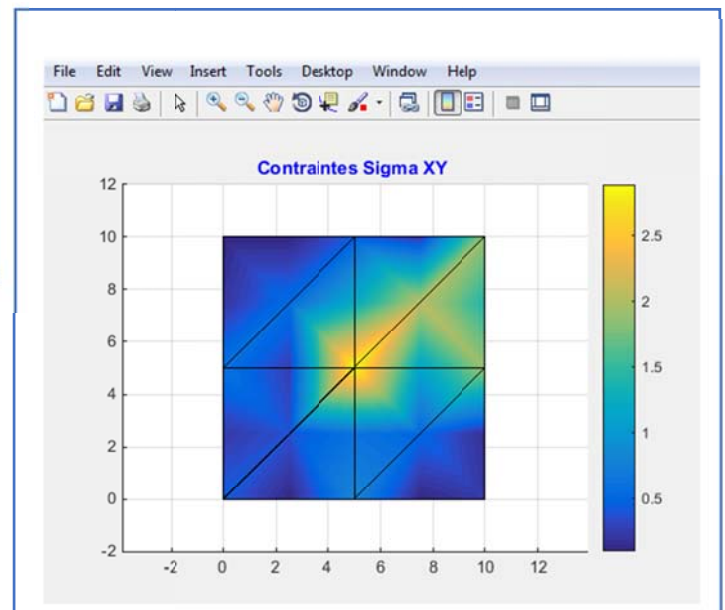


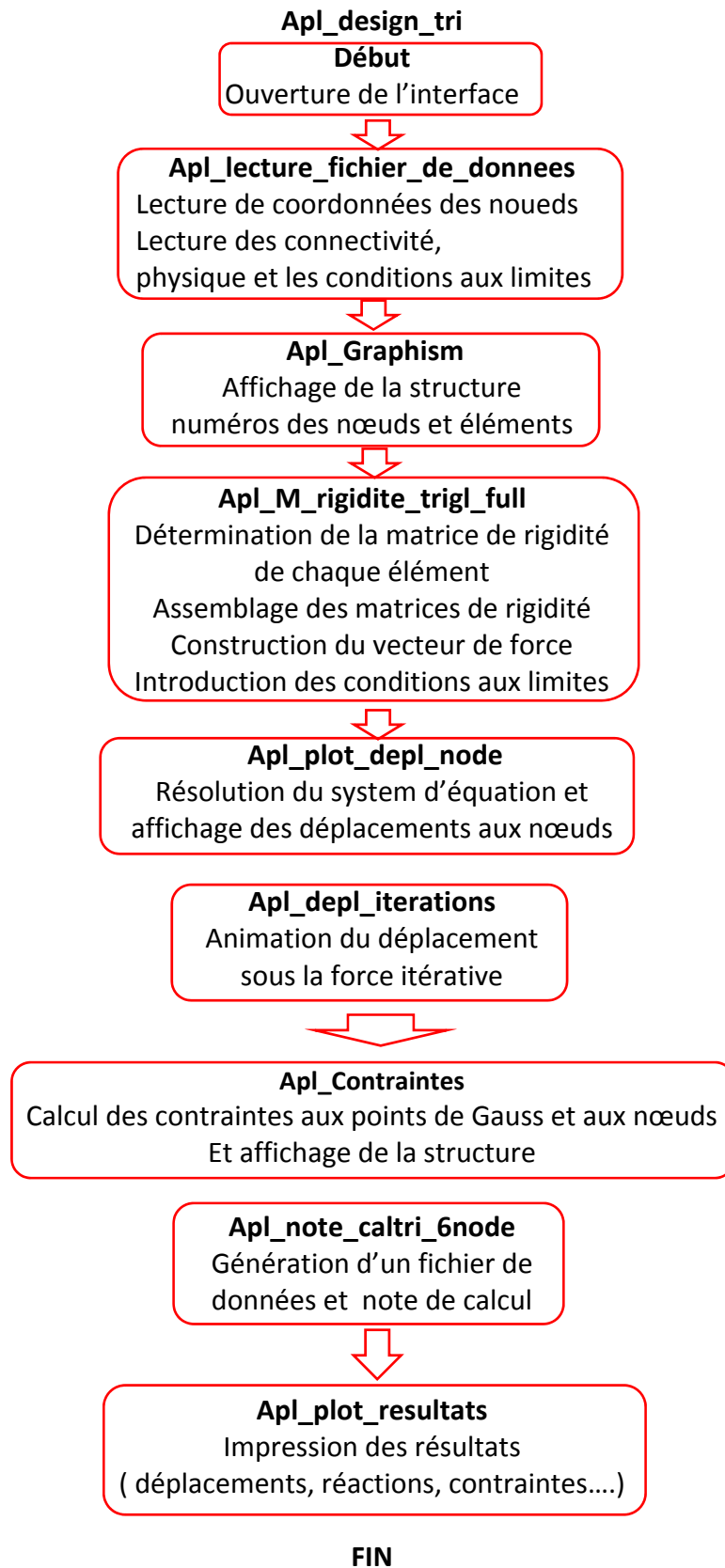
Figure 44 : Contrainte τ_{xy}

3.8.2 Sous programmes

Sous programmes	Fonction
affich_matrice.m	Affichage Matrice de rigidité
Apl_condition_limit.m	Affichage des conditions aux limites
Apl_Contraintes.m	Module contraintes
Apl_depl_iterations.m	Calcul des déplacements en itération
Apl_design_tri.m	Programme principal
Apl_DN_trigl.m	Construction de la matrice de dérivation des fonction de forme
Apl_Fct_Form_trigl.m	Fonctions de formes
Apl_matrice_D.m	Matrice D
Apl_Graphism.m	Dessin de la structure
Apl_lecture_fichier_de_donnees.m	Lecture des données
Apl_M_rigidite_trigl_full.m	Calcul de la matrice de rigidité de l'élément triangulaire
Apl_Maillage.m	Générateur de maillage élément triangle T6
Apl_matrix_B.m	Construction de la matrice B
Apl_note_caltri_6node.m	Elaboration de la note de calcul
Apl_plot_depl_node.m	Affichage des déplacements aux nœuds
Apl_plot_resultats.m	Affichage des tableaux des résultats
Apl_txt_elems_plot.m	Affichage des numéros des éléments
Apl_txt_node_plot.m	Affichage des numéros des nœuds

Tableau 8 : Sous-programmes Apl_Design_triangle T6

3.8.3 Organigramme de calculs



3.9 Apl Élément quadratique iso-paramétrique Q8

3.9.1 Interface du programme

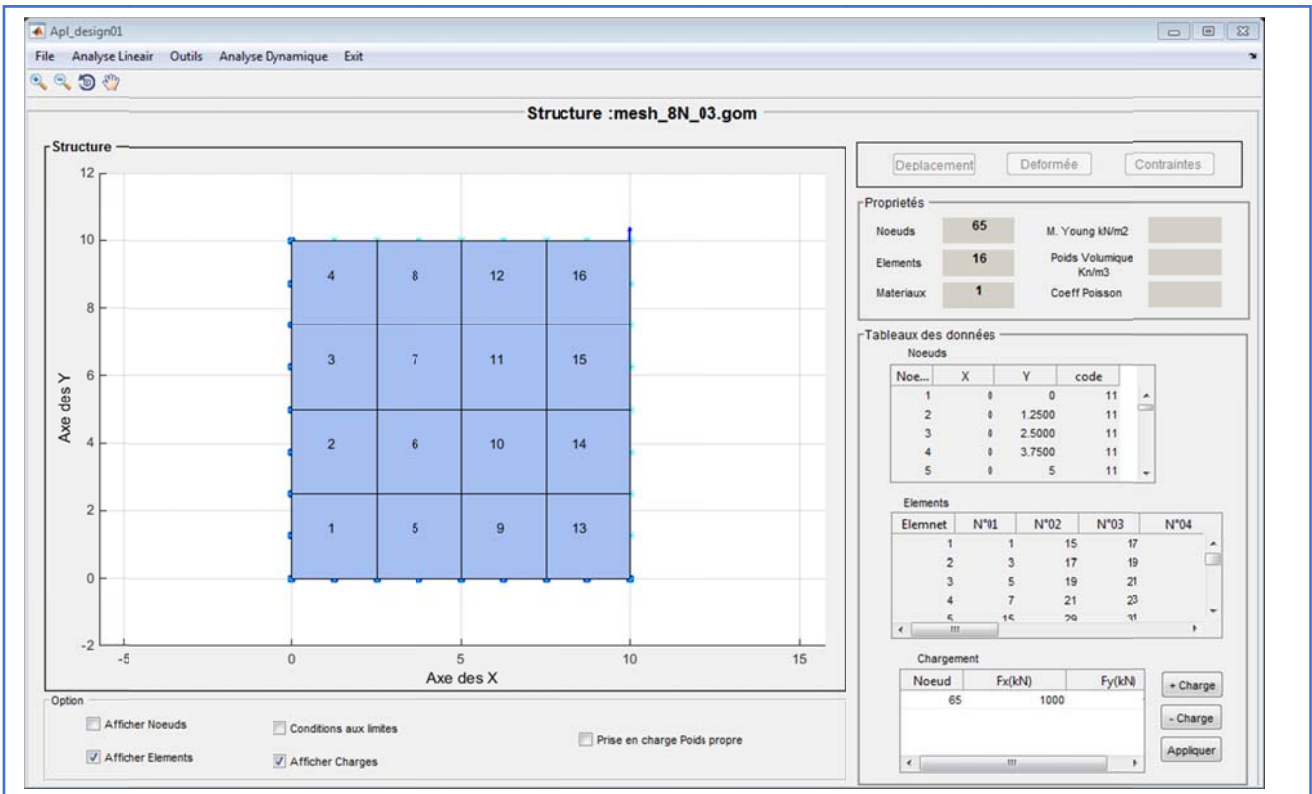


Figure 45 : interface de Apl_design_Q8

interface du module déplacements

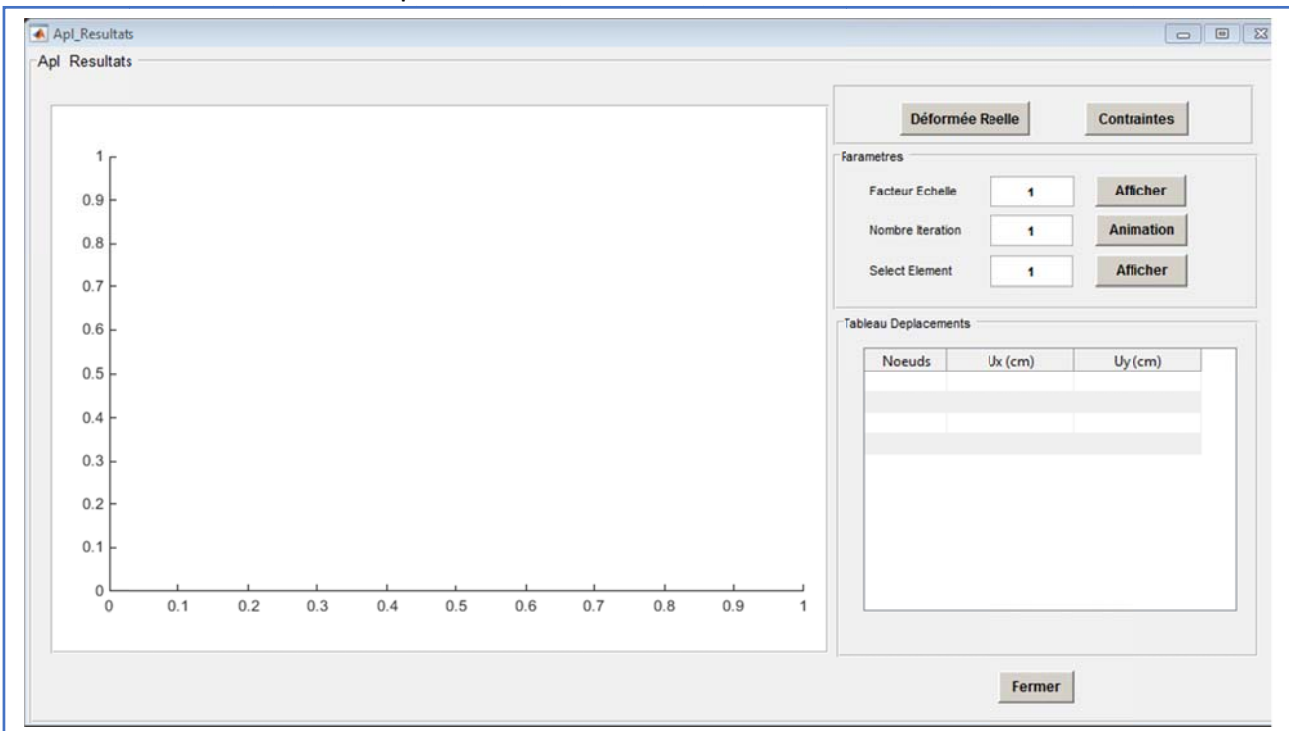


Figure 46 : interface du module déplacements

interface du Module contraintes

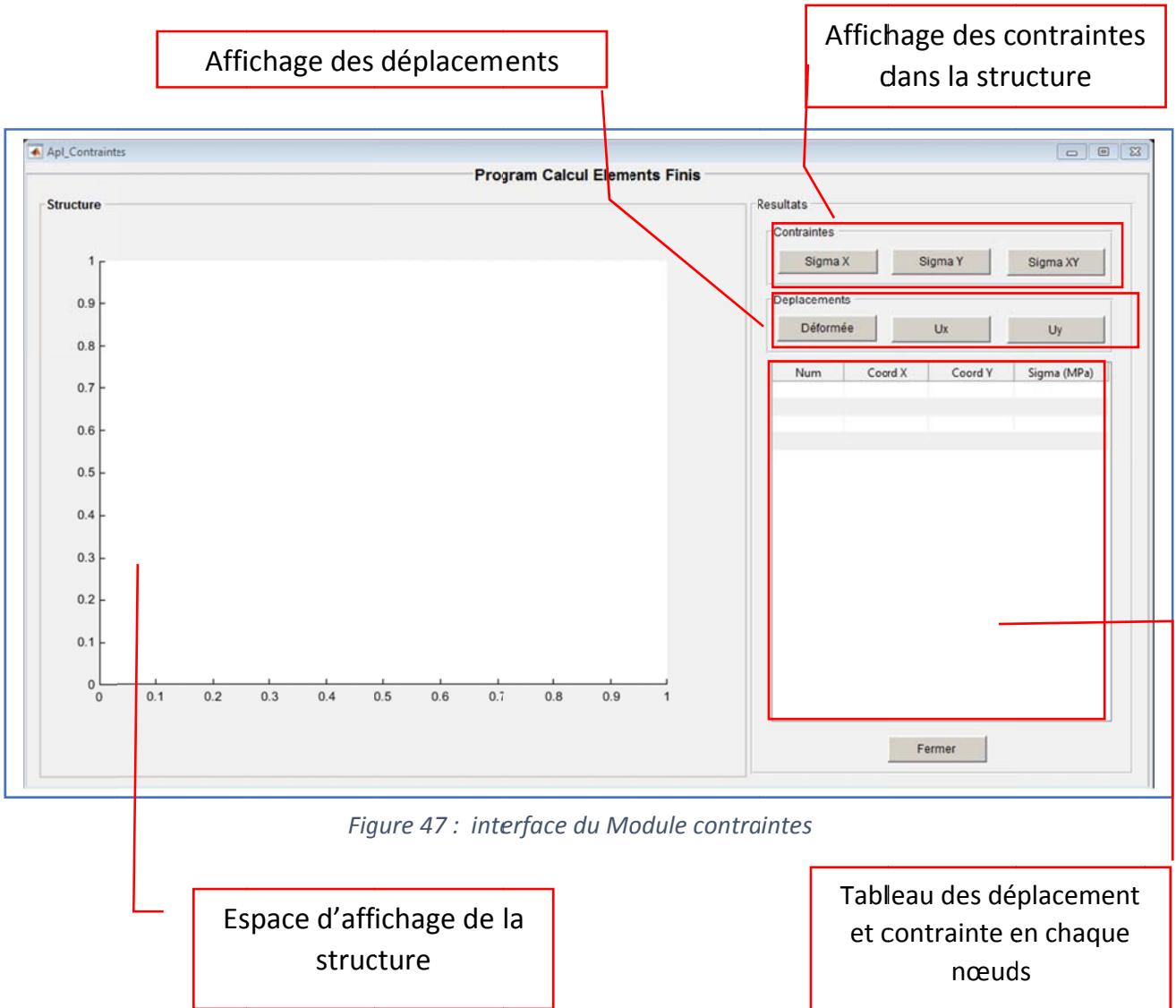


Figure 47 : interface du Module contraintes

Ou il est possible suivant un facteur d'agrandissement de visualiser la déformée U_x et U_y de la structure et les contraintes σ_x , σ_y et σ_{xy}

Dans l'interface graphique il est possible de visualisé un élément de la structure voir sa déformée ainsi que les contraintes aux différents nœuds de l'élément et aux points de Gauss le concernant

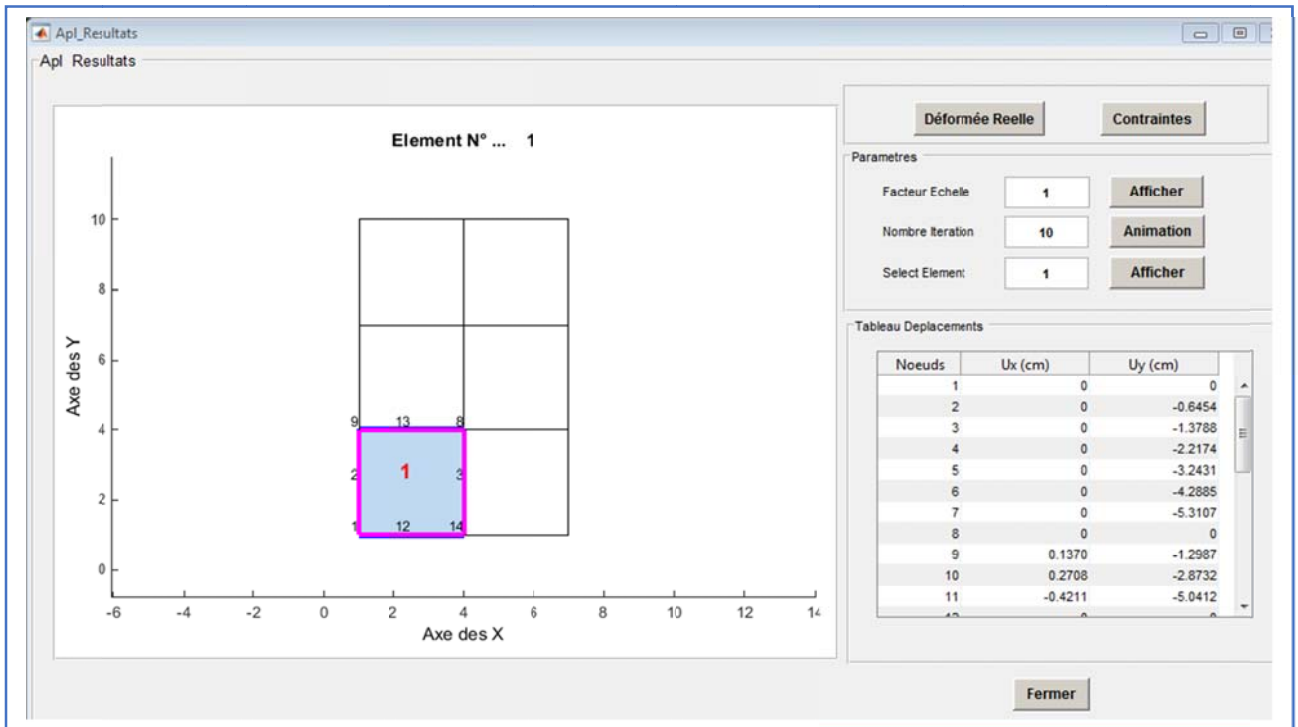


Figure 48 affichage d'un seul élément avec surbrillance son déplacement

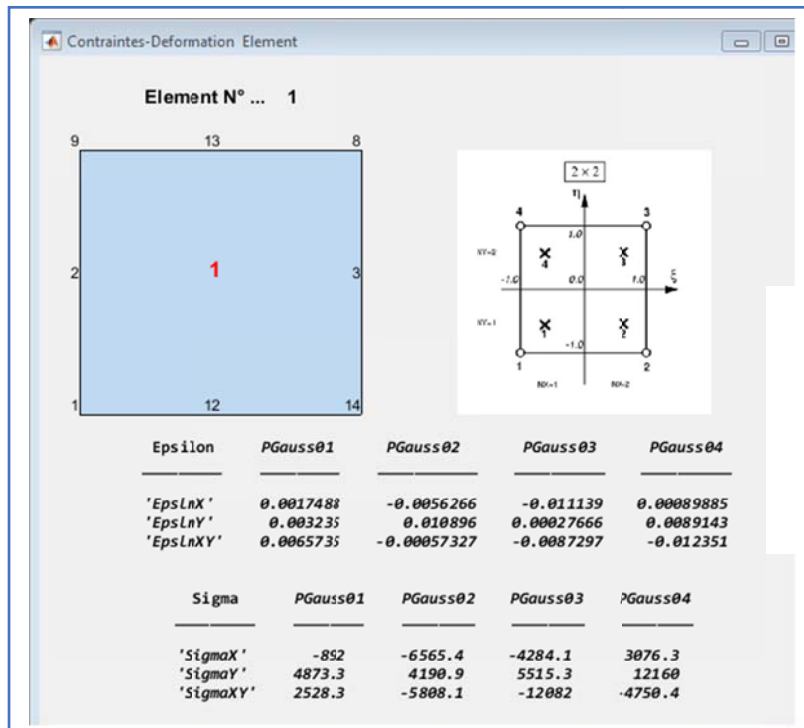


Figure 49 affichage des contrainte et déformation aux point de Gauss de l'élément

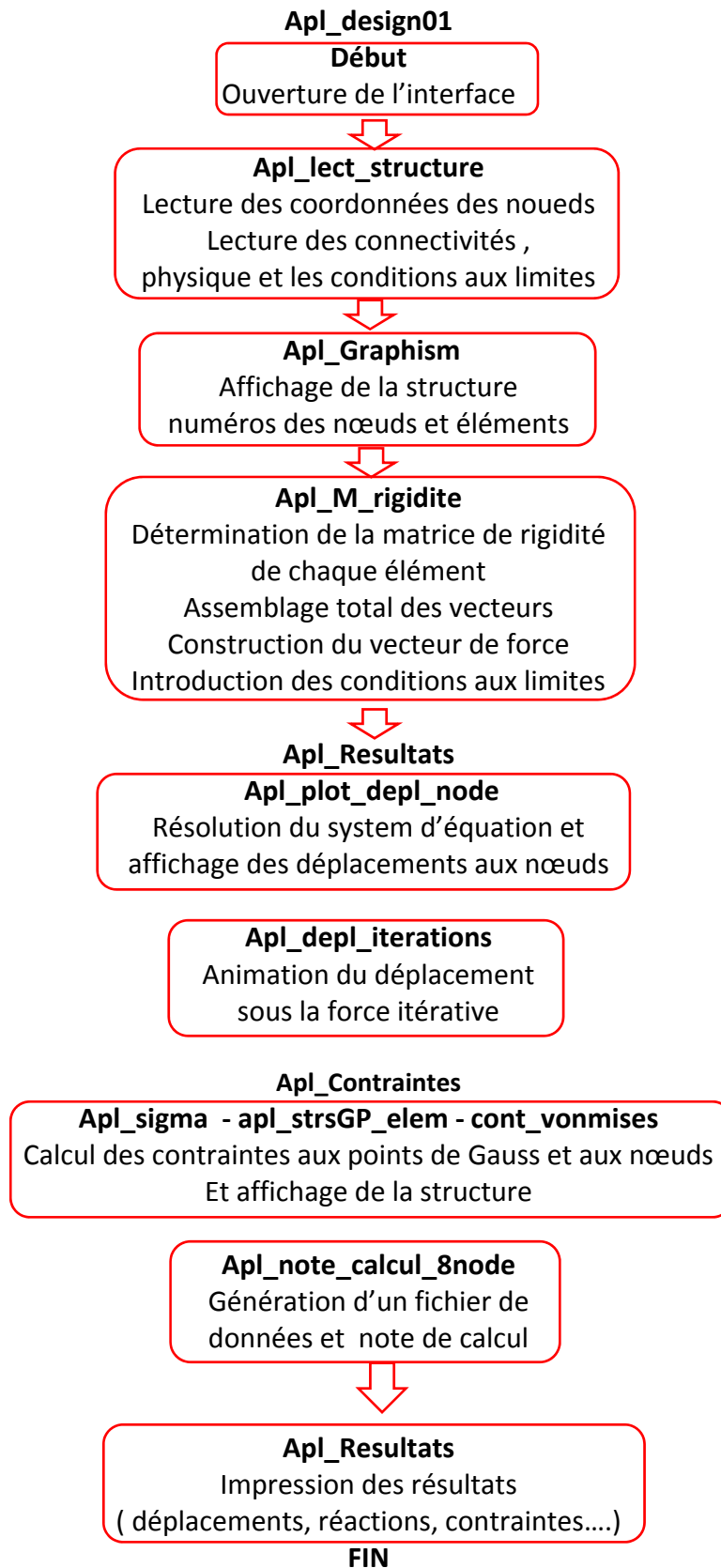
3.9.2 Sous programmes

Les différentes Sous programmes sont comme suit

Sous programmes	Fonction
Apl_design01.m	Programme principale
Apl_axes.m	Limitation des axes
Apl_lect_structure.m	Lecture des données
Apl_condition_limit.m	Affichage des conditions aux limites
Apl_node_plot.m	Dessin des nœuds de la structure
Apl_elems_plot.m	Dessin des numéros des éléments de la structure
Apl_plot_charg_node.m	Dessin des charges au nœuds
Apl_txt_elems_plot.m	Dessin des numéros des éléments
Apl_txt_node_plot.m	Dessin des numéros de nœuds
Apl_Graphism.m	Affichage structure
Apl_M_rigidite.m	Calcul de la matrice de rigidité
Apl_Fonct_Forme.m	Calcul des fonctions de forme
Apl_matrix_B.m	Construction de la matrice B
Apl_matrice_D.m	Matrice D
Apl_DNDRS.m	Calcul des dérivées des fonctions de formes
Apl_Resultats.m	Module de déplacements
Apl_plot_depl_node.m	Dessin des déplacements aux nœuds
Apl_depl_iterations.m	Implémentation des déplacements en fonction du vecteur de force itératif
Apl_Contraintes.m	Module contrainte
Apl_sigma.m	Implémentation des contraintes
Apl_constraint_iteration.m	Implémentation des contraintes en fonction du vecteur de force itératif
apl_strsGP_elem.m	Calcul des contraintes aux points de Gauss
Apl_table_figure.m	Visualisation des résultats sous forme de tableau
cont_vonmises.m	Calcul des contrainte de von mises
Apl_note_calcul_8node.m	Elaboration de la note de calculs
Apl_Dynamique.m	Calcul dynamique
Apl_Maillage.m	Module générateur de maillage élément quadratique iso a huit nœuds

Tableau 9 :Subroutines Apl_Design_Quadratique Q8

3.9.3 Organigramme de résolution



3.10 Apl Élément tétraédrique linéaire (T4)

3.10.1 Interface du programme

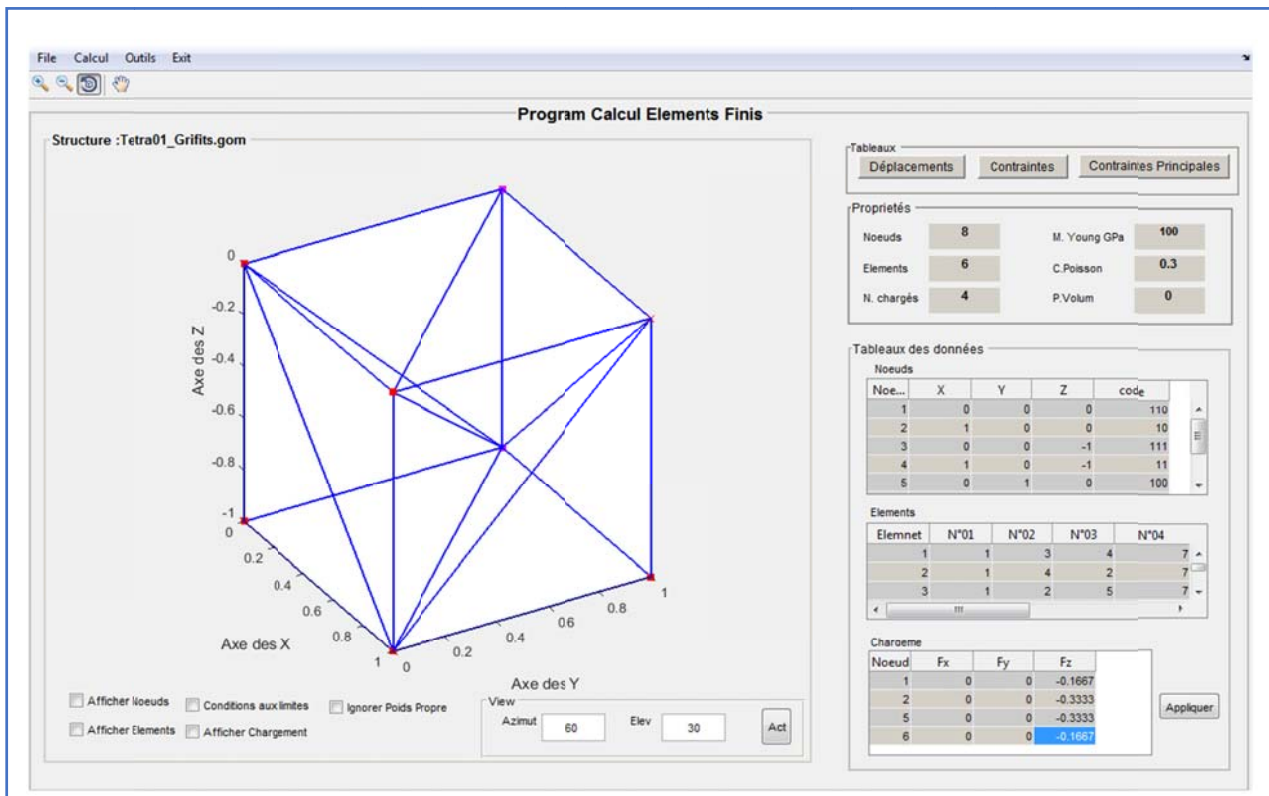


Figure 50 : interface de Apl_Design_tetraedre

Interface de visualisation des déplacements réel ou suivant un facteur d'agrandissement pour toute la structure ou pour chaque élément de tétraèdre de manière séparée.

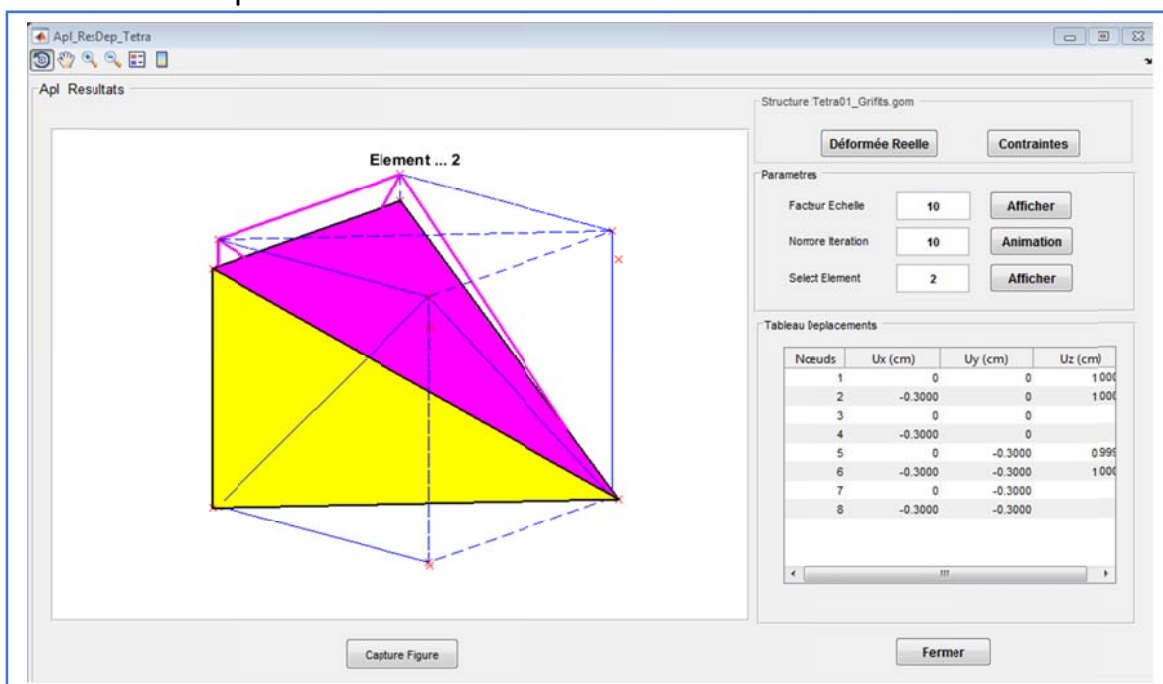


Figure 51 : interface du Module déplacements

L'interface des contraintes permet de visualiser la déformée et les contraintes ainsi que les contraintes principales aux nœuds et en chaque point de Gauss .

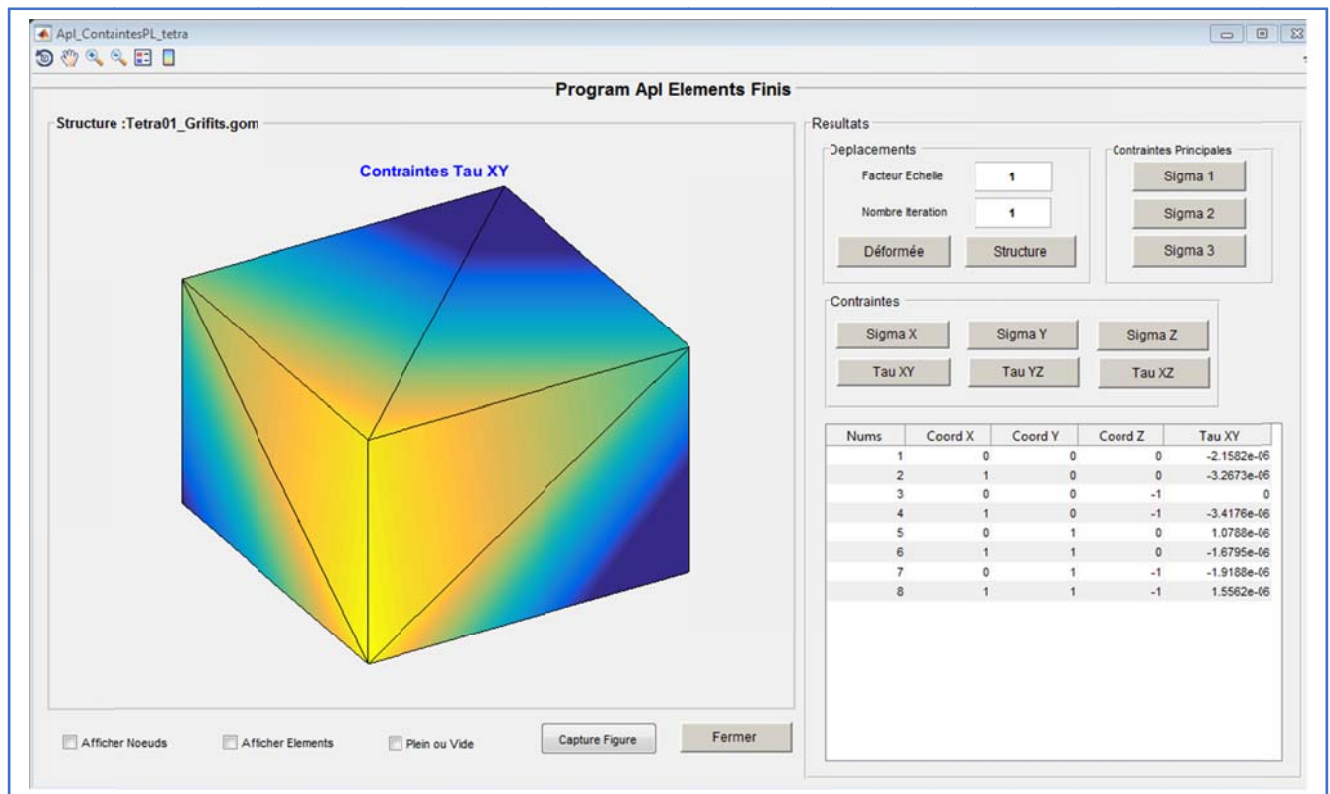


Figure 52 : interface du Module contraintes

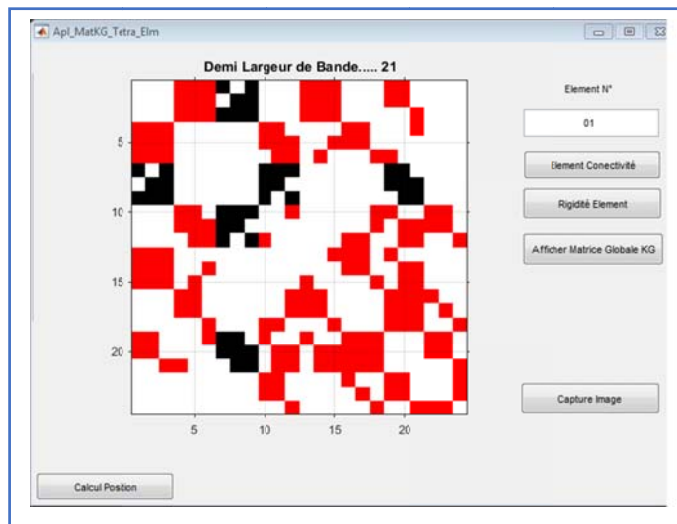


Figure 53 : Affichage matrice de rigidité

la matrice de rigidité global Et la position de la matrice de rigidité de chaque élément dans le repère global

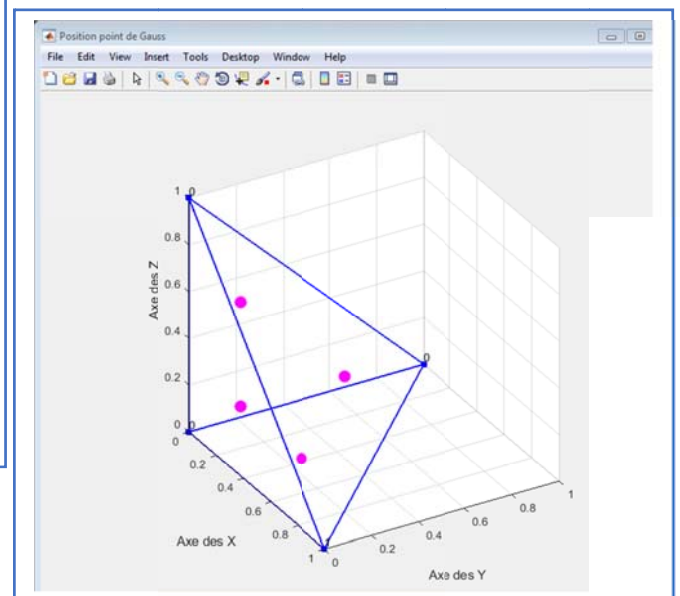


Figure 54 : Affichage des points de Gauss

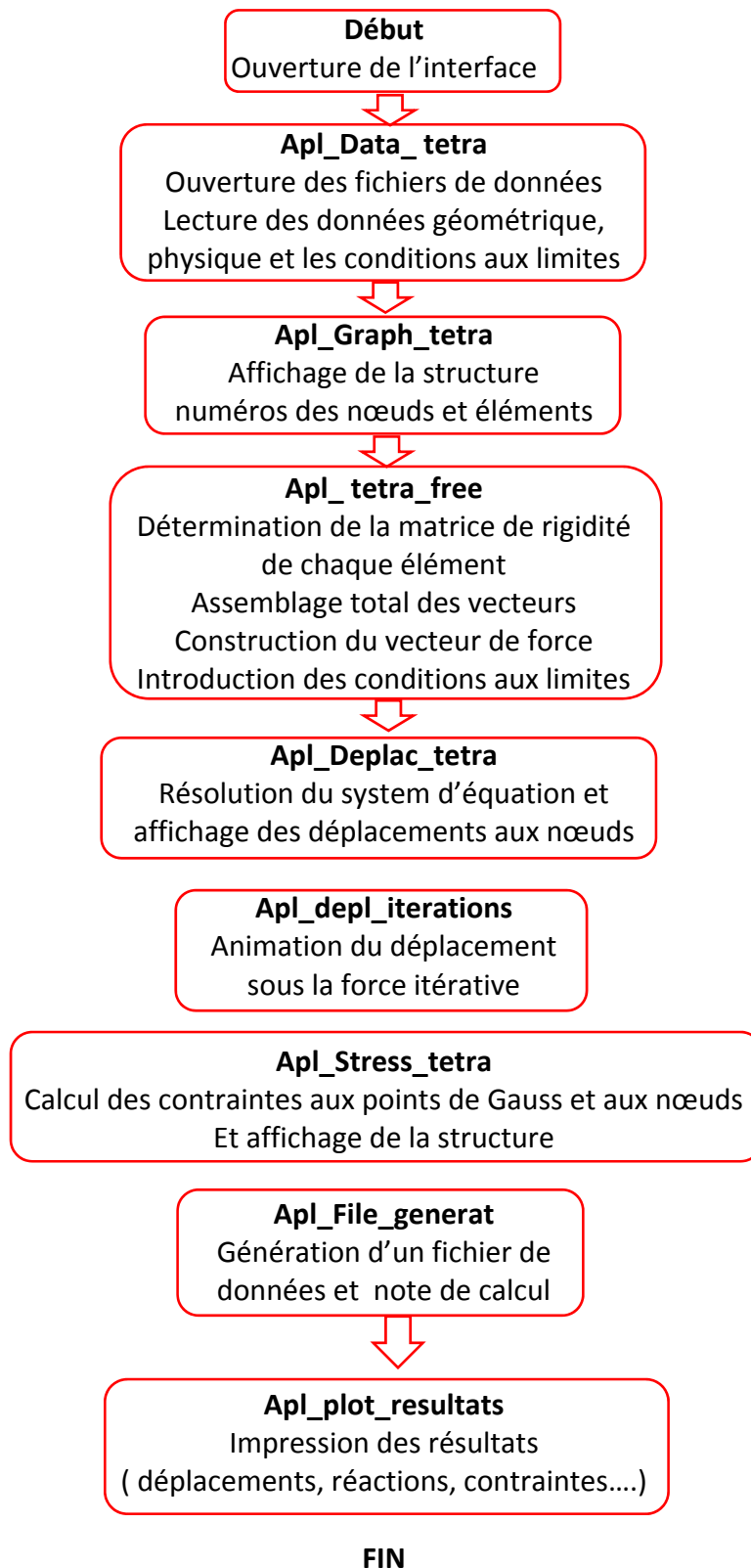
Possibilité de visualiser la position des points de Gauss pour l'élément tétraèdre

3.10.2 Sous programmes

Sous programmes	Fonction
Apl_Tetra_free	Programme principale
Apl_Data_Tetra	Lecture du fichier des données coordonnées connectivités et chargements
Apl_Graph_tetra	Affichage structure
Apl_vecXY	Construction du vecteur des coordonnées de chaque élément
Apl_Points_Gauss	Construction de la matrice des points de Gauss
Apl_dNr	Construction des matrice de forme
Apl_Der_FoncForm	Calcul des dérivées des fonctions de formes
Apl_BJdet	Construction de la matrice B
Apl_condition_limit	Affichage des conditions aux limites
Apl_Deplac_tetra	Implémentation des déplacements
Apl_Stress_tetra	Implémentation des contraintes
Apl_depl_iterations	Implémentation des déplacements en fonction du vecteur de force itératif
Apl_matrice_D.m	Matrice D
Apl_plot_resultats	Visualisation des résultats sous forme de tableau
Apl_File_generat	Générateur de fichier de données
Show_Cube_PG	Visualisation des points de Gauss
Apl_mesh_tetra	Module générateur de maillage élément Cubique a huit noeuds

Tableau 10 : Soubrogrammes Apl_Design_Tétraedre

3.10.3 Organigramme de résolution



3.11 Apl Élément brique linéaire (solide C8)

3.11.1 Interface du programme

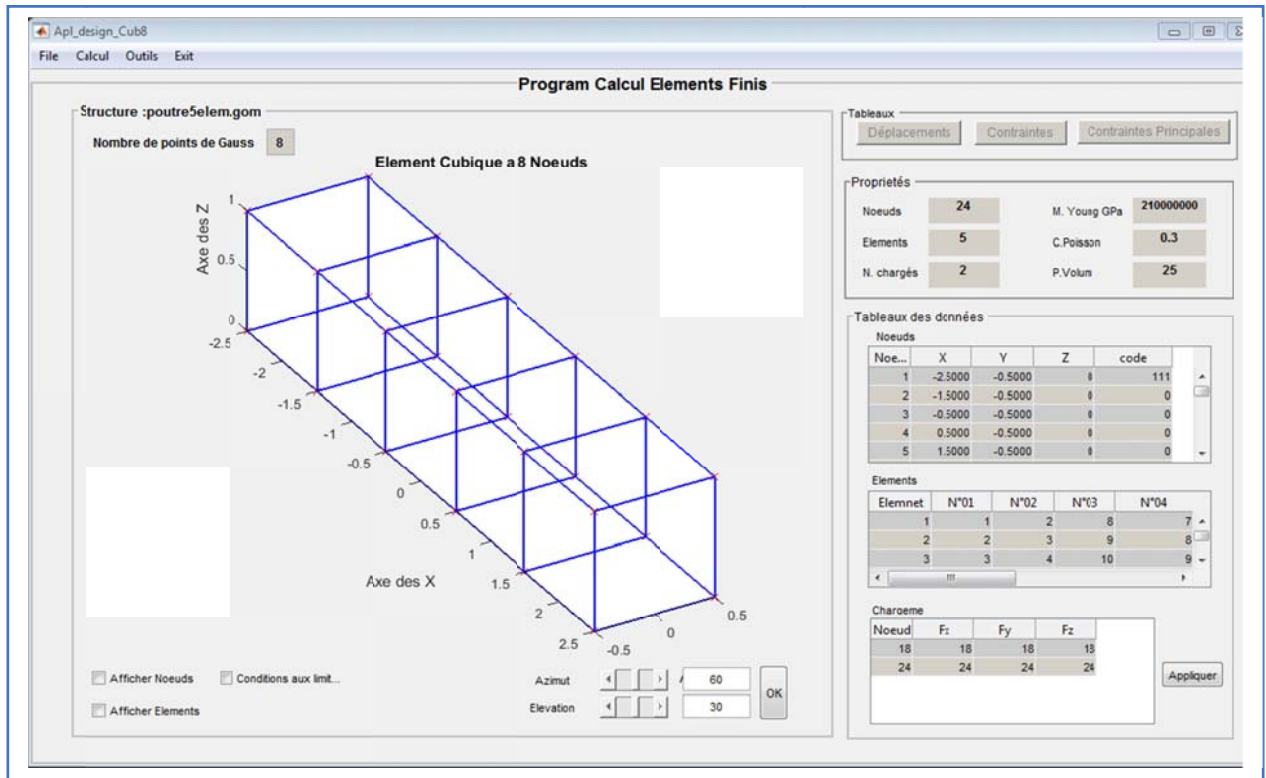


Figure 55 Interface Apl_Design Cub8

Les déplacements en chaque nœuds sont donnés sous forme de tableau après exécution des calculs

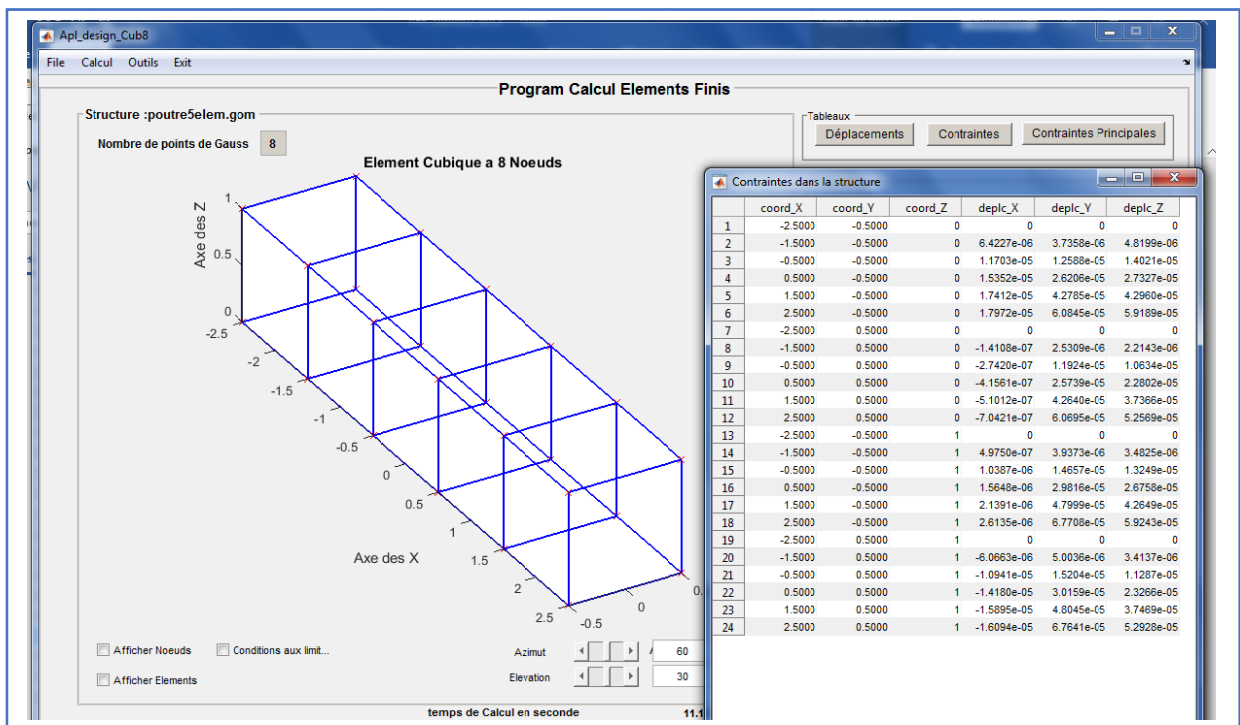


Figure 56 Affichage des déplacements

Sous l'Interface graphique il est possible de visualiser les déplacements réel ou avec un facteur d'échelle ainsi que les déplacements à chaque itération

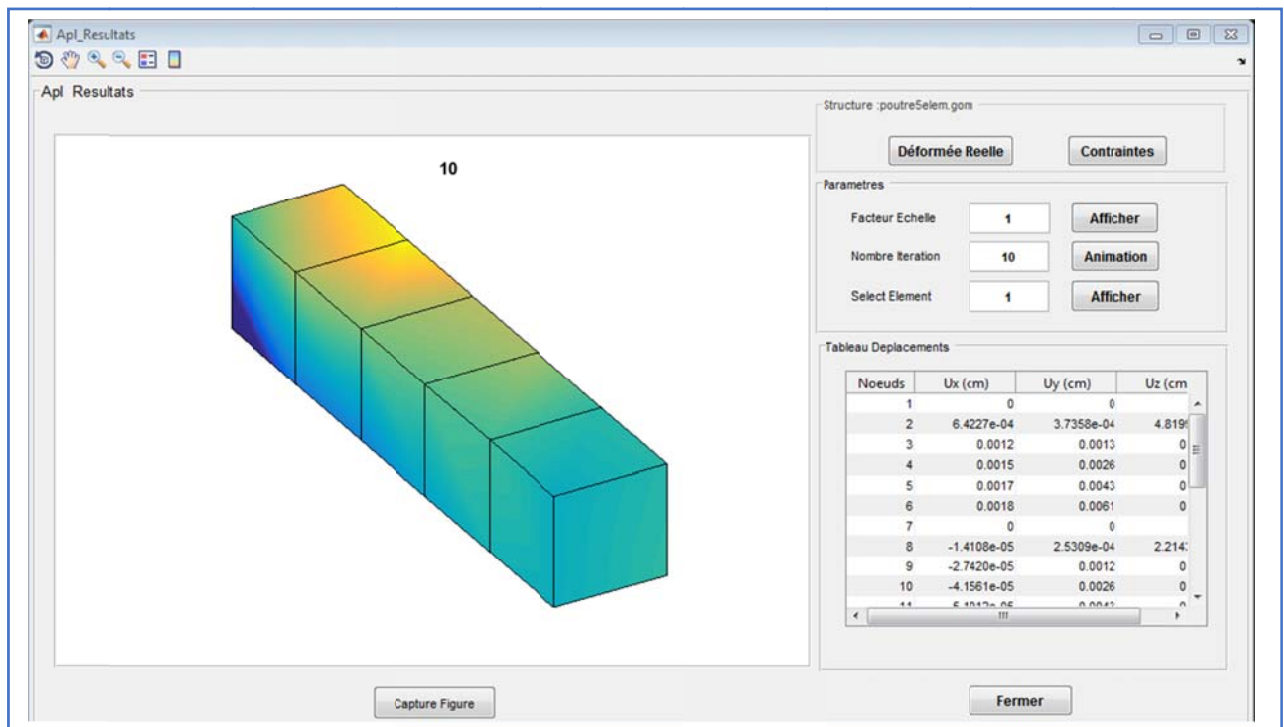


Figure 57 Interface des déplacements

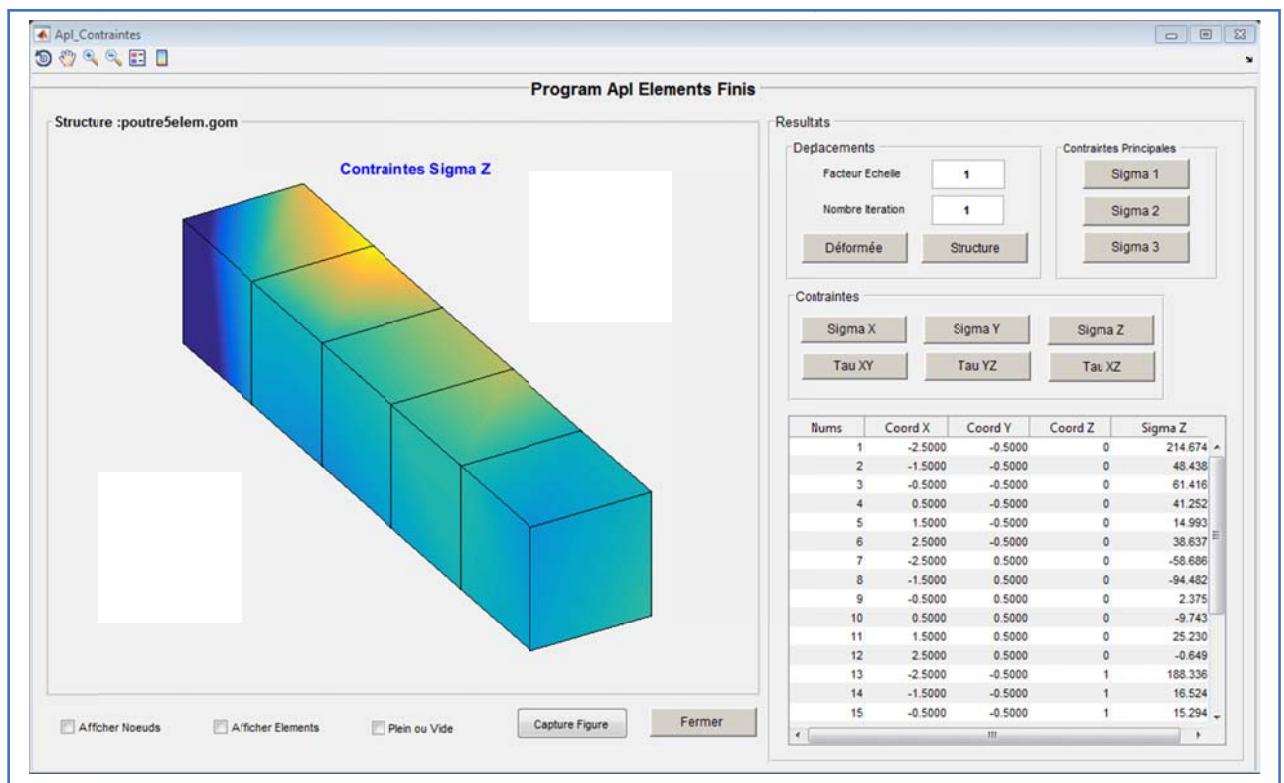


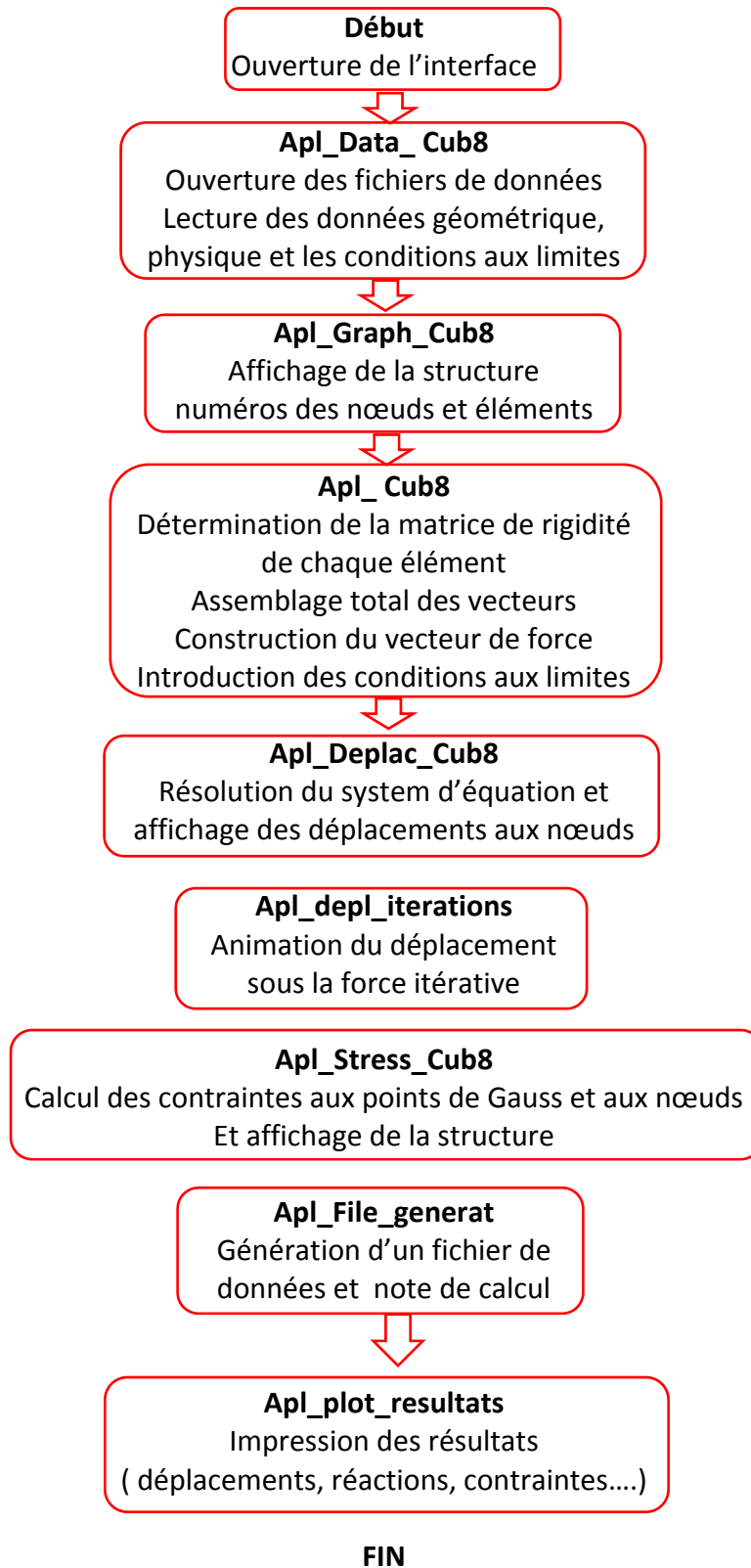
Figure 58 Interface des contraintes

3.11.2 Sous Programmes

Sous programmes	Fonction
Apl_cub8_free	Programme principale
Apl_Data_cub8	Lecture du fichier des données coordonnées connectivités et chargements
Apl_Graph_cub8	Affichage structure
Apl_vecXY	Construction du vecteur des coordonnées de chaque élément
Apl_Points_Gauss	Construction de la matrice des points de Gauss
Apl_dNr	Construction des matrice de forme
Apl_Der_FoncForm	Calcul des dérivées des fonctions de formes
Apl_BJdet	Construction de la matrice B
Apl_condition_limit	Affichage des conditions aux limites
Apl_Deplac_cub8	Implémentation des déplacements
Apl_Stress_cub8	Implémentation des contraintes
Apl_matrice_D.m	Matrice D
Apl_depl_iterations	Implémentation des déplacements en fonction du vecteur de force itératif
Apl_plot_resultats	Visualisation des résultats sous forme de tableau
Apl_File_generat	Générateur de fichier de données
Show_Cube_PG	Visualisation des points de Gauss
Apl_mesh_cub8	Module générateur de maillage élément Cubique a huit noeuds

Tableau 11 :Sous programmes Apl_Design_Cub8

3.11.3 Organigramme de résolution



3.12 Apl Élément tétraédron (C20)

3.12.1 Interface du programme

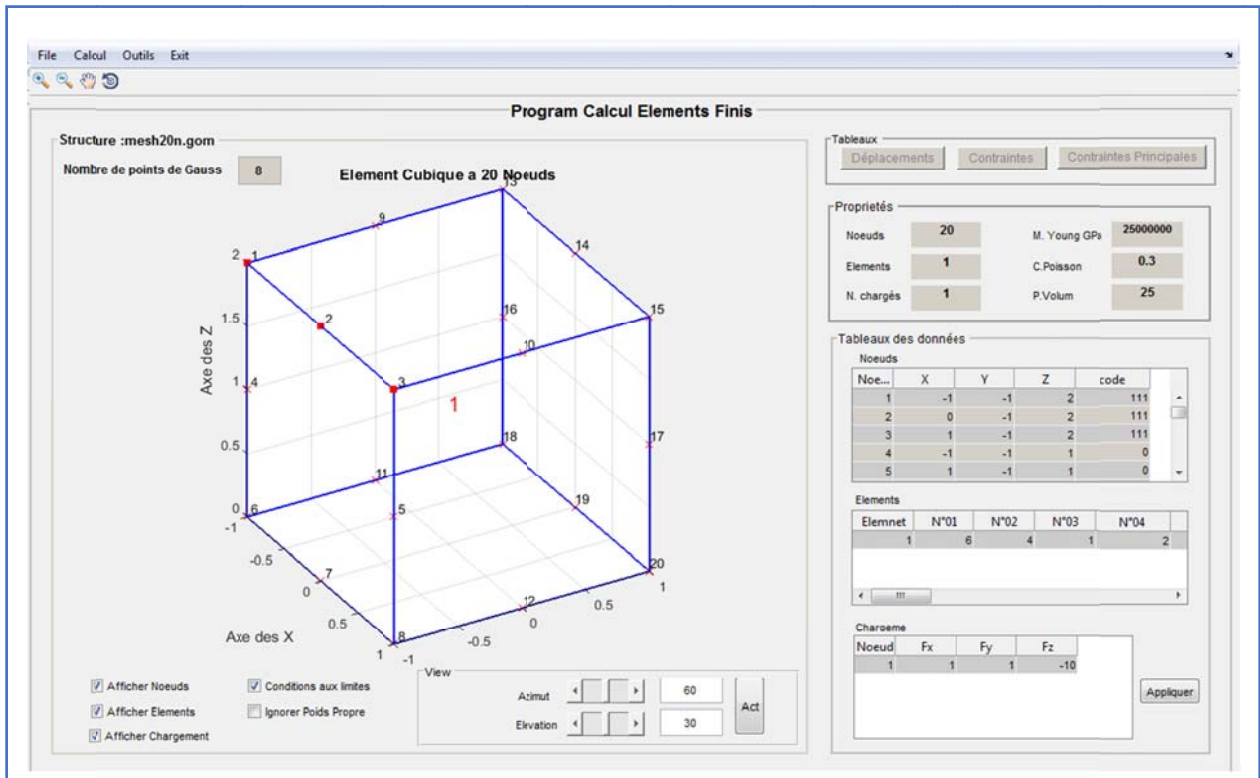


Figure 59 :interface Apl_Design_Cub20

Et lancement des calculs affichage du module des déplacements

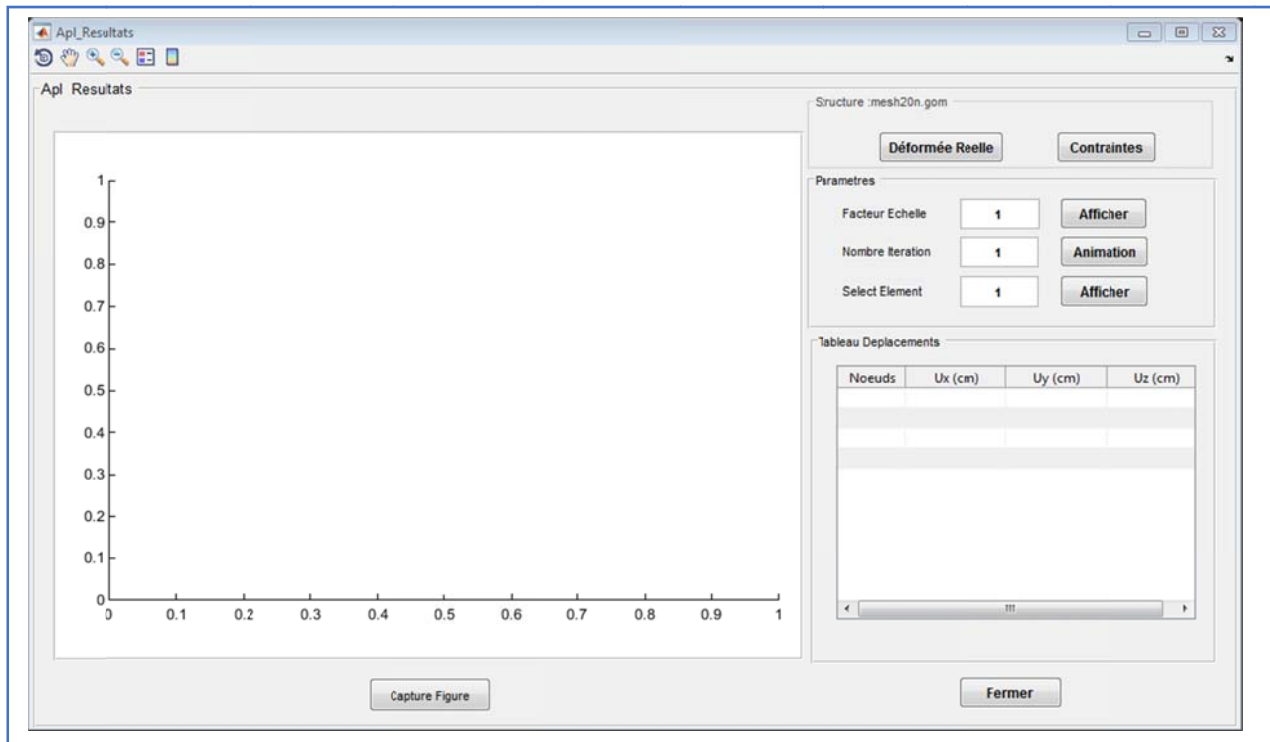


Figure 60 : Interface des déplacements

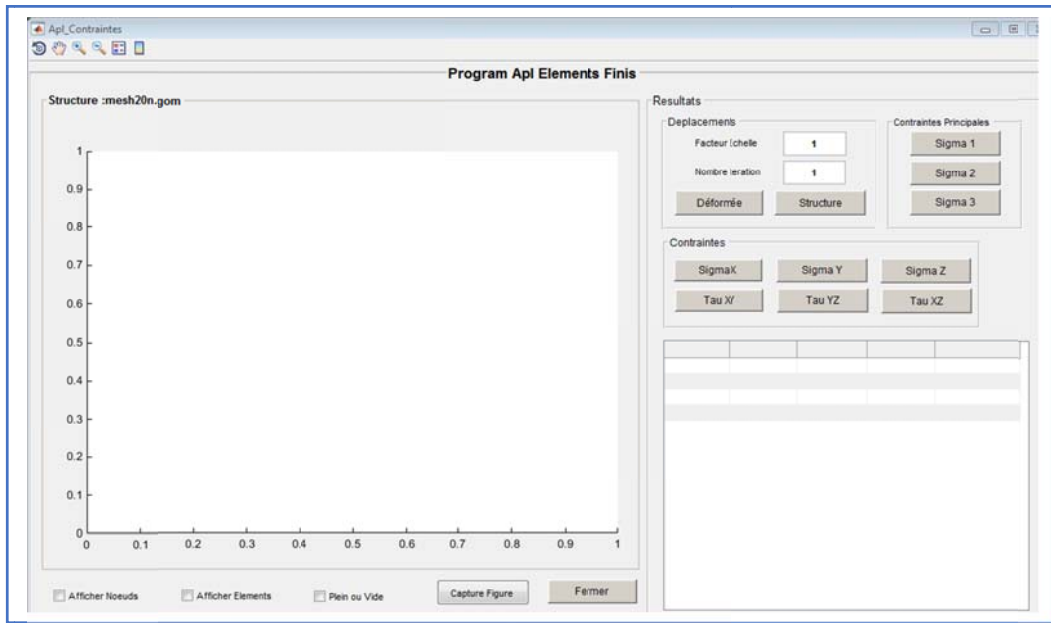


Figure 61 :Interface des contraintes

Visualisation de la matrice de rigidité Globale et la demie largeur de bande

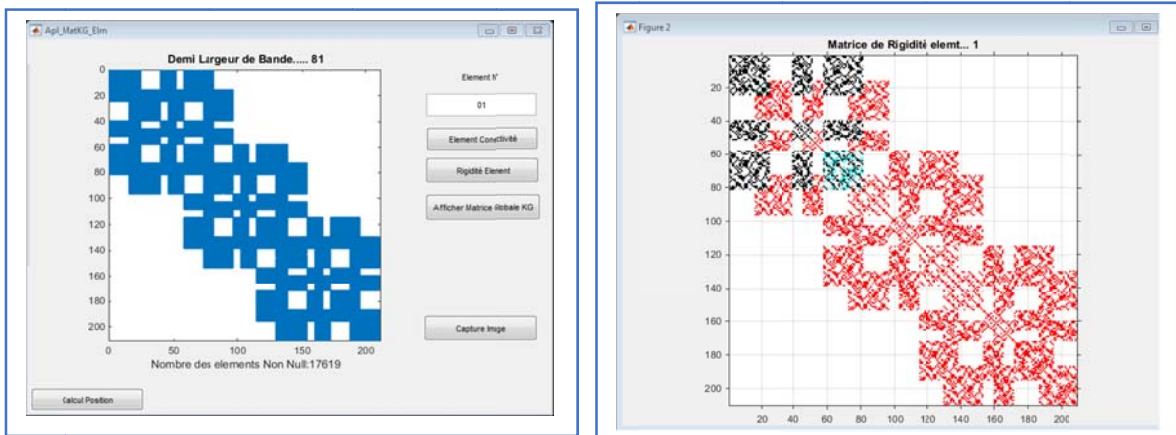


Figure 62 :Affichage matrice de rigidité globale

Visualisation des points de Gauss de l'éléments

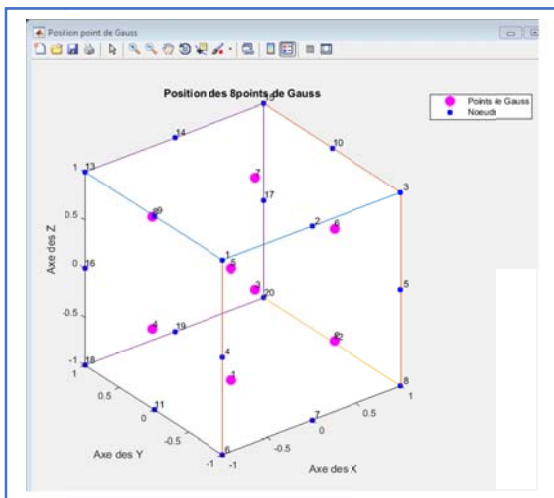


Figure 63 :Affiche 8 Points de Gauss

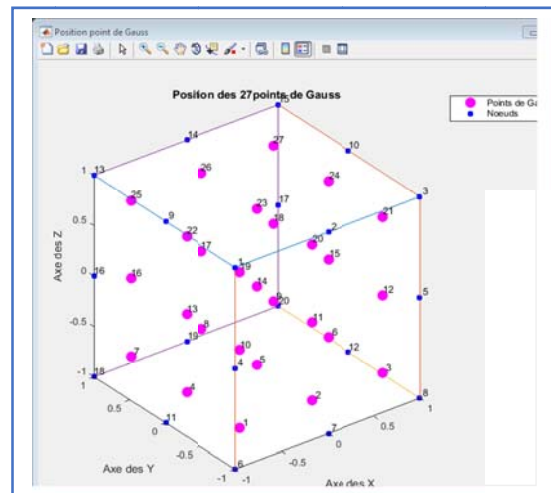


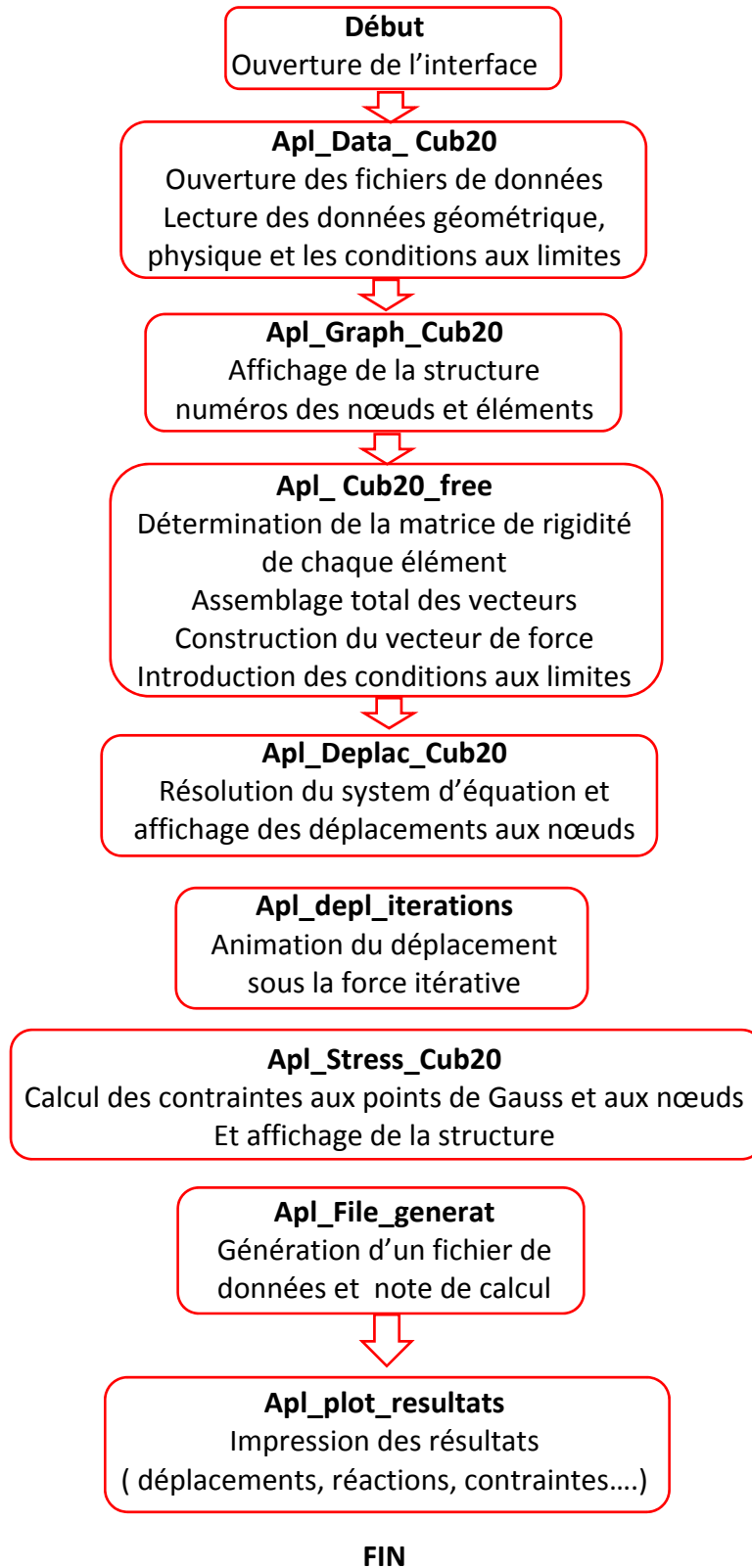
Figure 64 :Affiche 27 Points de Gauss

3.12.2 Sous programme

Sous programmes	Fonction
Apl_Cub20_free	Programme principale
Apl_Data_Cub20	Lecture du fichier des données coordonnées connectivités et chargements
Apl_Graph_Cub20	Affichage de la structure
Apl_vecXY	Construction du vecteur des coordonnées de chaque élément
Apl_Points_Gauss	Construction de la matrice des points de Gauss
Apl_dNr	Construction de la matrices des fonctions de formes
Apl_Der_FoncForm	Calcul des dérivées des fonctions de formes
Apl_BJdet	Construction de la matrice B
Apl_matrice_D.m	Matrice D
Apl_condition_limit	Affichage des conditions aux limites
Apl_Deplac_Cub20	Implémentation des déplacements
Apl_Stress_Cub20	Implémentation des contraintes
Apl_depl_iterations	Implémentation des déplacements en fonction du vecteur de force itératif
Apl_plot_resultats	Visualisation des résultats sous forme de tableau
Apl_File_generat	Générateur de fichier de données
Show_Cube20_PG	Visualisation des points de Gauss
Apl_mesh_Cub20	Module générateur de maillage élément Cubique a huit noeuds

Tableau 12 : Subroutines Apl_Design_Cub20

3.12.3 Organigramme de résolution



Chapitre 4

Validation du Programme

Apl_Design MARS Version1.0

Sommaire

4	Introduction.....	132
4.1	Introduction	132
4.2	Apl design Barre	133
4.3	Apl design Poutre	137
4.4	Apl design Q8	139
4.5	Apl design T6	147
4.6	Apl design Tetra4.....	153
4.7	Apl design Cub8.....	166
4.8	Apl design Hexa20	158

4.1 Introduction

La validation du programme **Apl_Design MARS** 1.0, se fera pour chaque module et par des modèles réduits classiques et simples afin d'évaluer l'écart entre les résultats obtenus avec les théories classiques en élasticité ou en R.D.M ou, ceux donnés avec le logiciel Autodesk Robot Structural Analysis Professional 2013, qui est un logiciel très puissant en calcul des structures.

4.2 Apl design Barre

Le premier exemple de calcul est une structure en treillis figure 63

La vérification a été effectuée avec le programme de structure ROBOT pour la comparaison des résultats obtenue avec **APL_DESIGN_MARS**

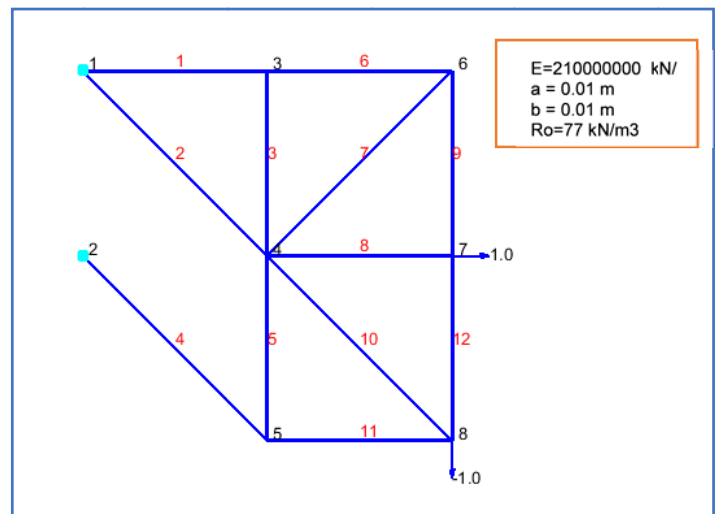


Figure 65 : exemple treillis

La figure ci-dessous montre l’affichage de La structure étudiée dans la zone graphique de notre programme, l’affichage des différents détails de la structure est optionnel (numéros des nœuds et des barres ,conditions aux limites ,positions et valeurs des chargements).

Options Affichage

- Charges appliquées aux noeuds Afficher
- Nombre de Barre 12 Afficher
- Nombre de Noeud 8 Afficher

Données de l'étude

Noeuds

Noeuds	X	Y	Code
1	0	6	11
2	0	3	11
3	3	6	0
4	3	3	0

Barres

Barre	N° 01	N° 02	a	b	E M.Young	Ro F
1	1	3	0.0100	0.0100	210000000	77.0
2	4	1	0.0100	0.0100	210000000	77.0
3	4	3	0.0100	0.0100	210000000	77.0

Charges

Noeud	Sens X	Sens Y
7	1	0
8	0	-1

Appliquer

Figure 66 : affichage exemple treillis dans le programme

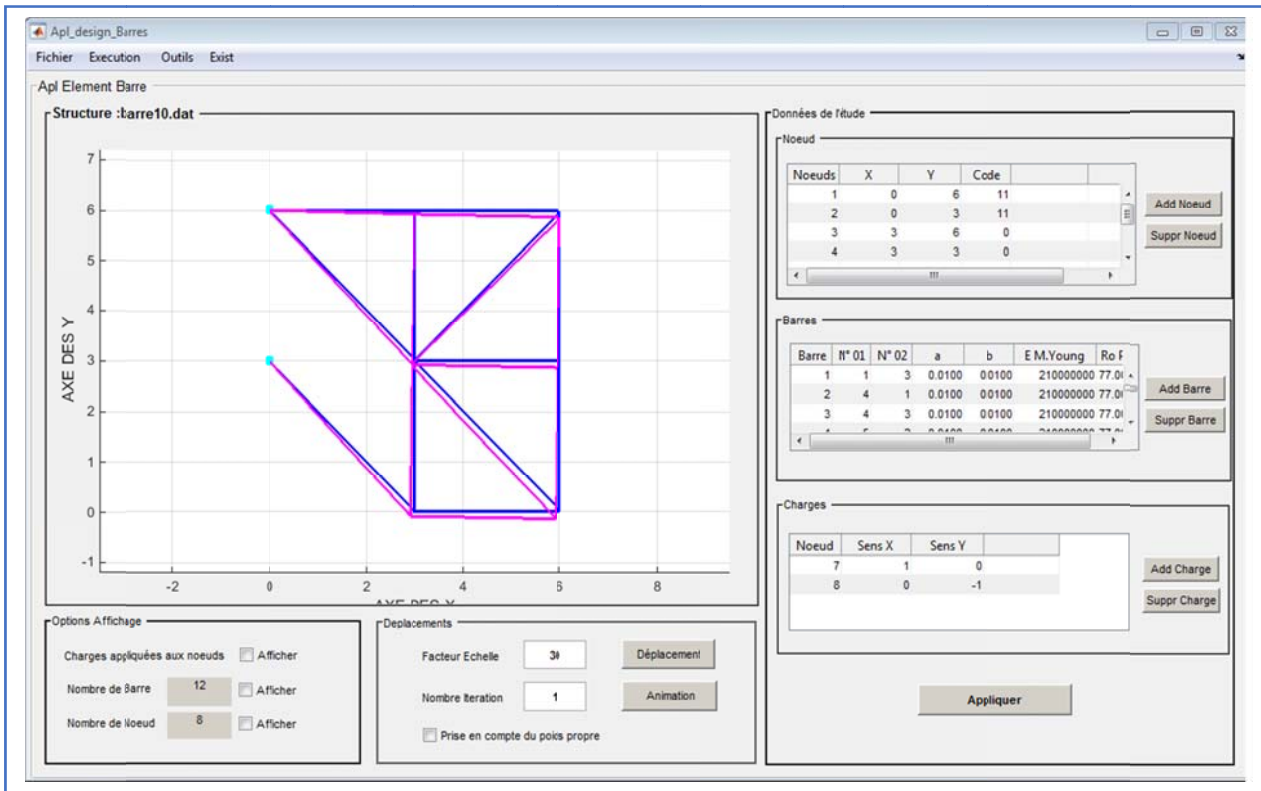


Figure 67 : déplacements dans la structure treillis

Les résultats de calcul sont donnés dans des tableaux avec déplacements aux nœuds, réaction d'appuis et les Effort Normal ainsi que les contraintes dans les barres , une note de calcul au forma txt est automatiquement générée.

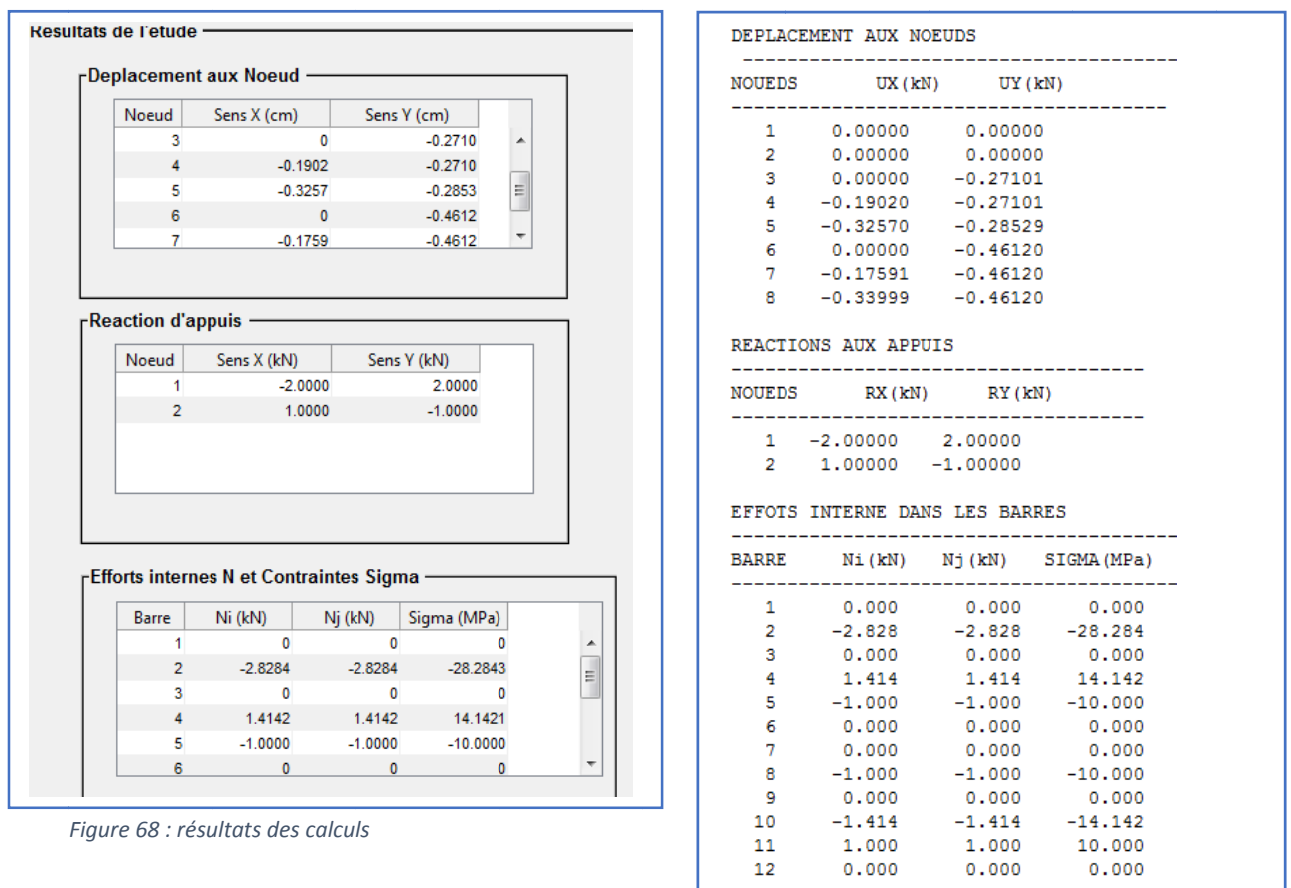


Figure 68 : résultats des calculs

Validation de l'exemple 01 et comparaison avec le logiciel Autodesk

Robot Structural Analysis Professional 2013

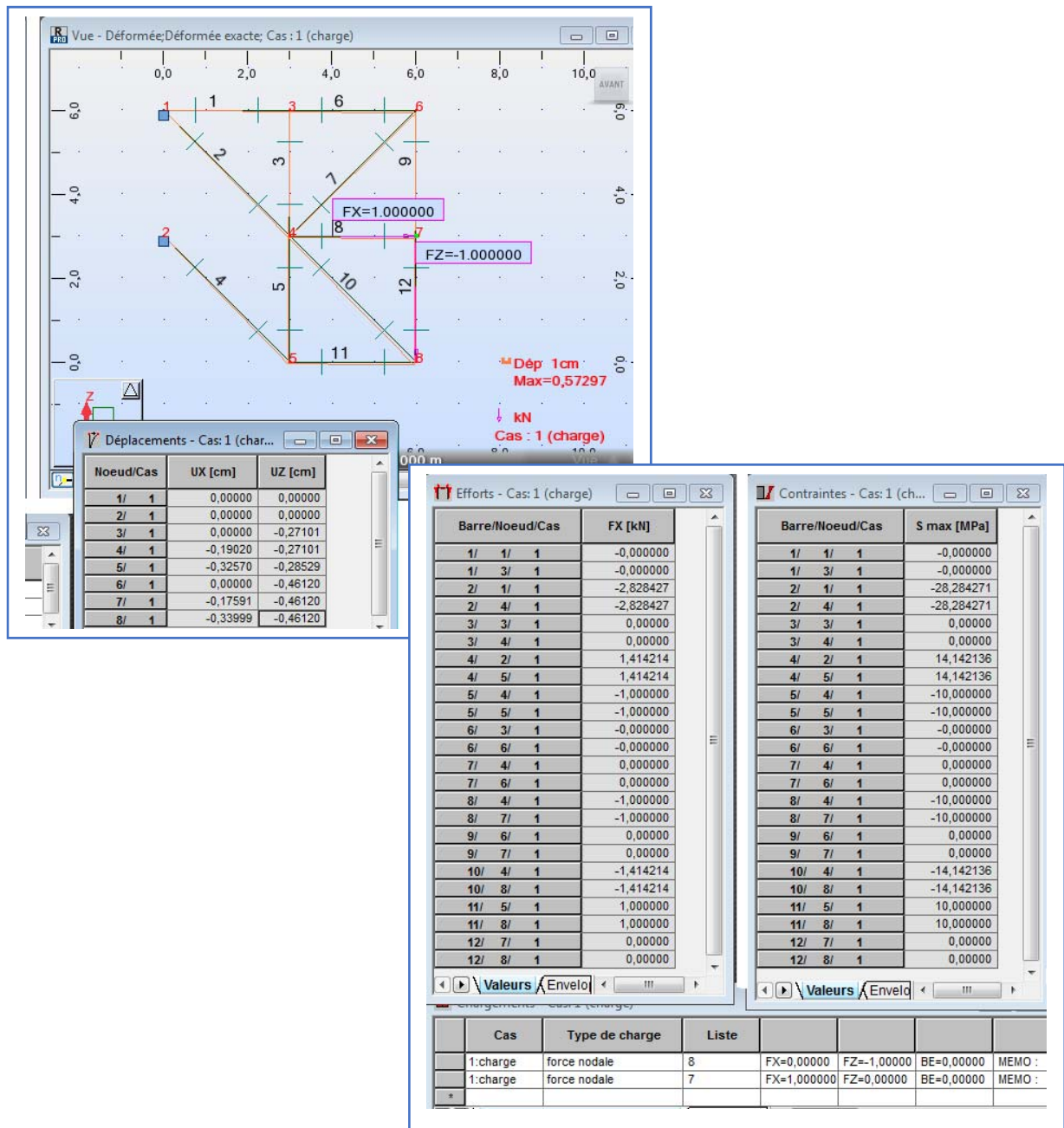


Figure 69 : résultats obtenus avec Autodesk Robot Structural Analysis Professional

la prise en compte du poids propre est possible de manière optionnel et les résultats obtenus sont comme suit ; figure 68

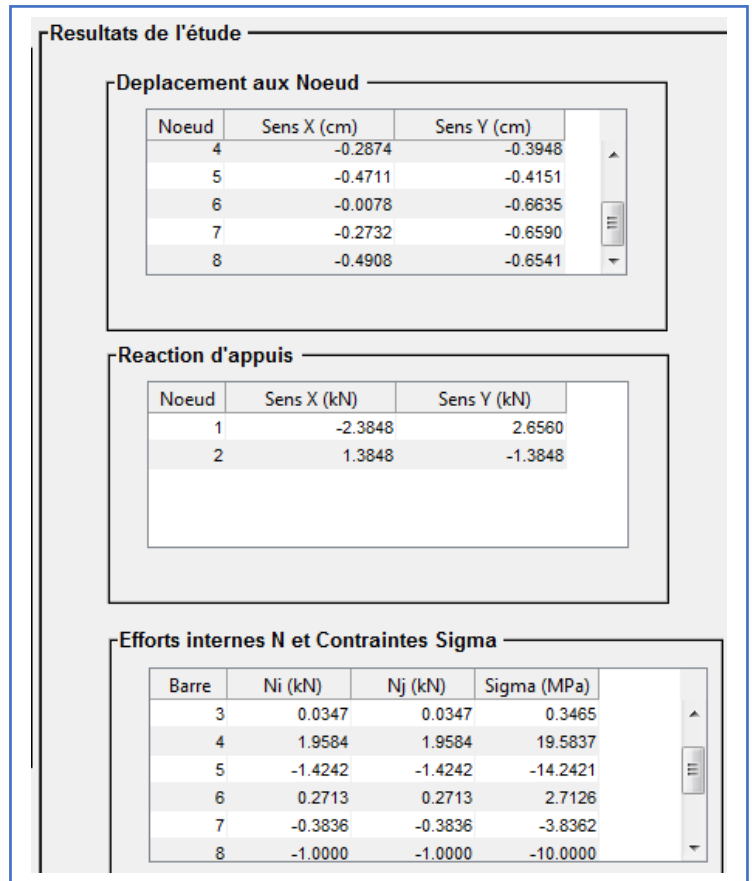


Figure 70 : Résultat obtenues avec prise en compte du poids propre

En comparaison avec le logiciel Autodesk Robot Structural

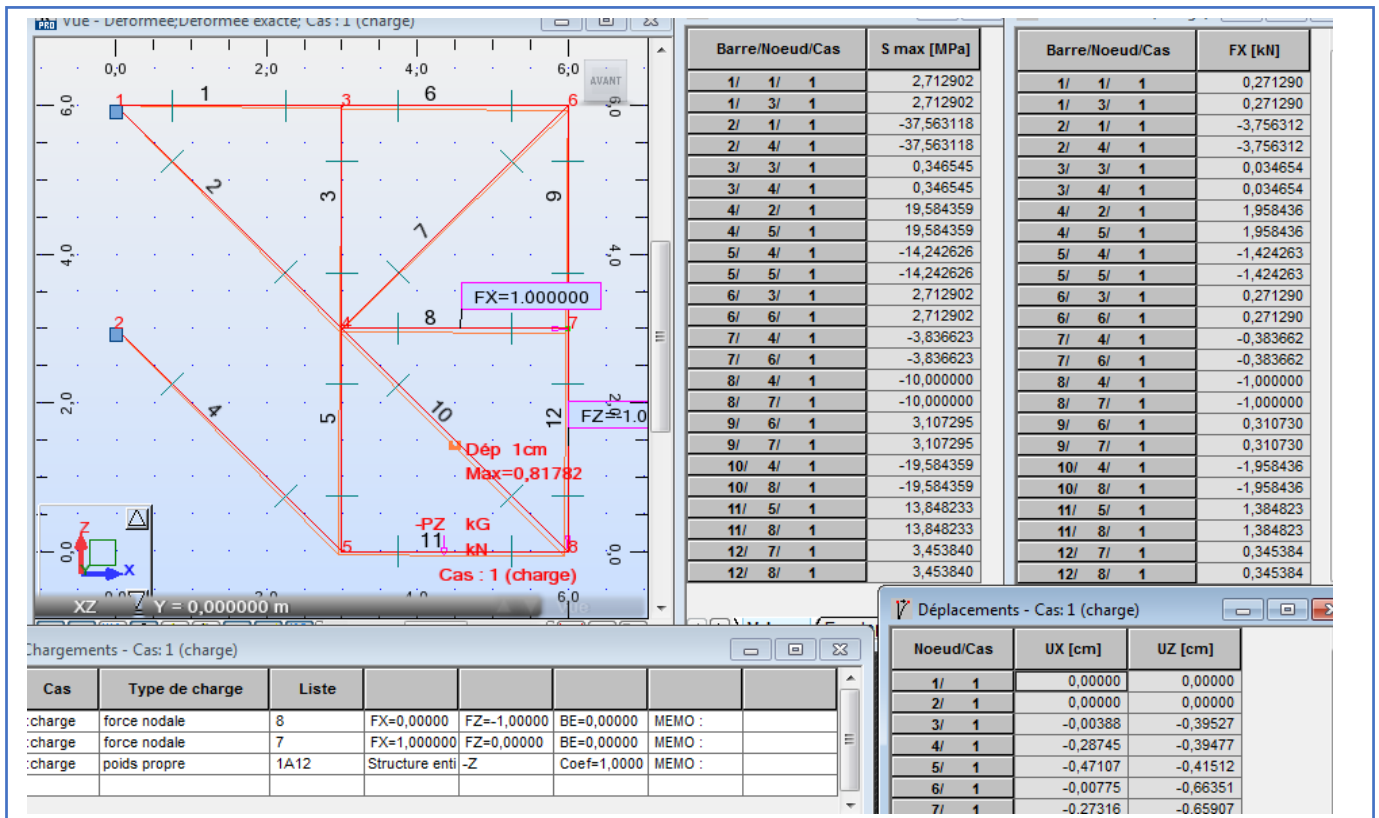


Figure 71 : Résultat obtenus avec Autodesk Robot Structural

4.3 Apl design Poutre

soit à étudier la structure suivante

$E = 210 \text{ GPa}$
 $I = 4.1666666667 \cdot 10^{-7} \text{ m}^4$
 $S = 2.0 \cdot 10^{-3} \text{ m}^2$
 $P = 2.0 \text{ kN}$
 $q_0 = 10.0 \text{ kN/m}$

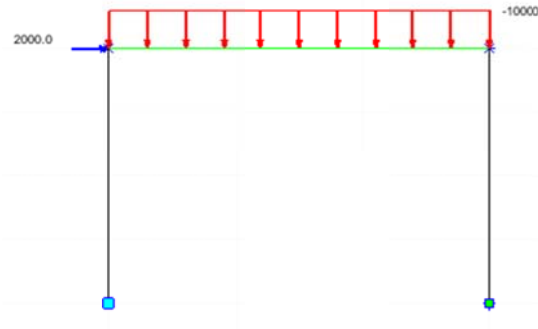


Figure 72 : exemple portique

Après ouverture du fichier de données et lancement des calculs

Structure : poutre010.pin

Données de l'étude

Noeud

Noe...	X	Y	Cod...	Cod...	Rot
1	0	0	111		
2	0	4	0		
3	6	4	0		
4	6	0	110		

Barres

Barre	N° 01	N° 02	a	b	M. Young	Coeff...
1	2	1	0.0020	4.1667...	2.1000e+11	77001
2	3	4	0.0020	4.1667...	2.1000e+11	77001
3	2	3	0.0020	4.1667...	2.1000e+11	77001

Charges

Noeud	Sens X	Sens Y	Rotation
2	2000	0	0

Barre	Type	Charg X	Charg Y	Dist a	Dist b
3	3	0	-10000	0	0

Element Poutre

Charges appliquées aux noeuds Afficher

Nombre de Barre Afficher

Nombre de Noeud Afficher

Charges

Facteur Echelle Affich

Nombre Iteration Animation

Appliquer

Figure 73 : affichage de la structure dans le programme Apl_design Poutre

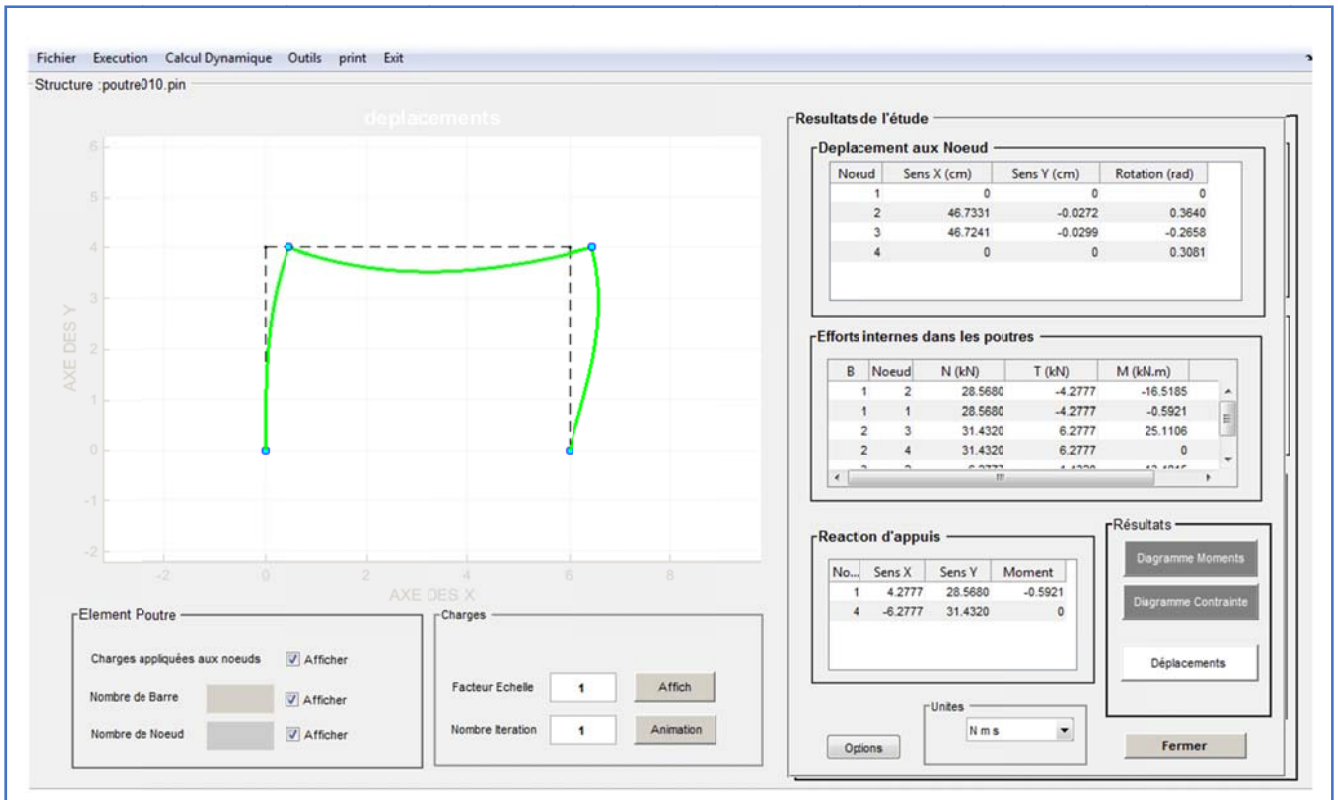


Figure 74 : Déformée de la structure

En comparaison avec le logiciel Autodesk Robot Structural

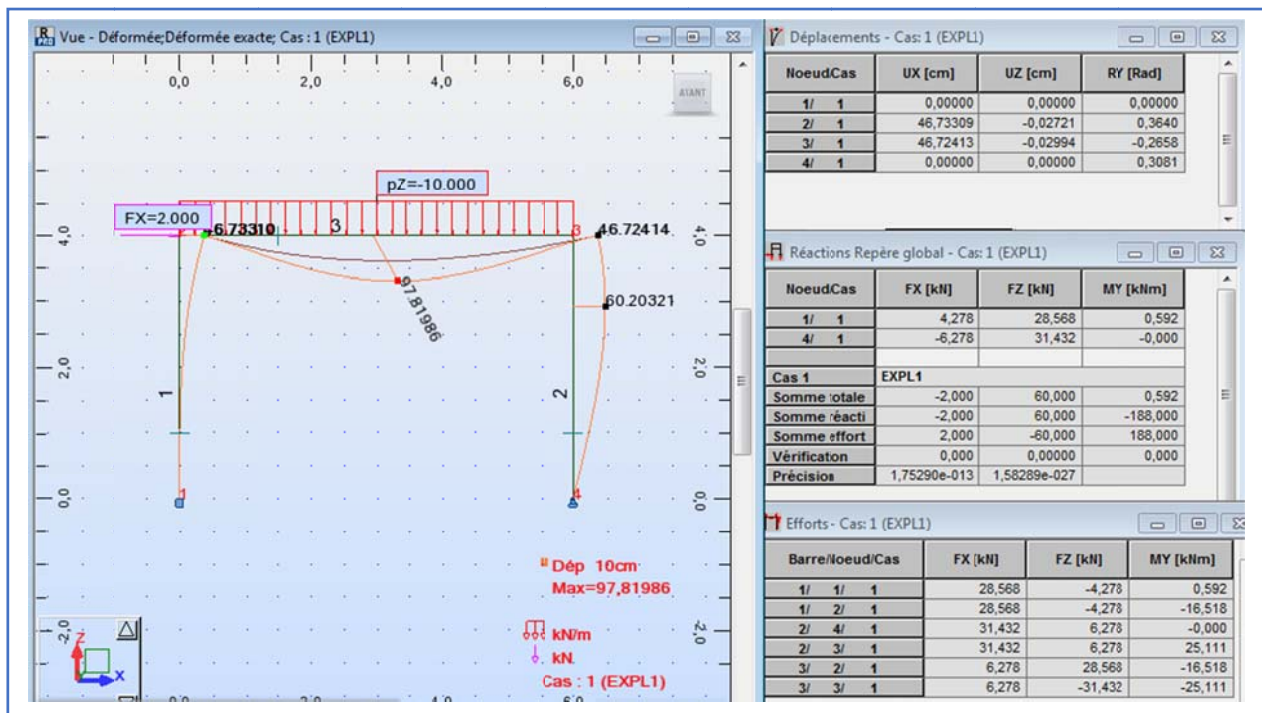


Figure 75 : Résultat obtenus par Autodesk Robot Structural Analysis Professional

4.4 Apl design Q8

Test 01

On se propose d'étudier la flexion d'une poutre encastree a l'une de ses extremités et libre de l'autre, soumise a une charge concentree $P=100$ kN, de longueur $L = 4.00$ m, de section rectangulaire $b = 0.4$ m, $h=0.5$ m, et sans tenir compte du poids propre de la structure.

$$E = 32\,000 \text{ MPa}$$

$$\nu = 0.21$$

$$P = 100 \text{ kN}$$

$$L = 4 \text{ m}$$

$$b = 0.4 \text{ m}$$

$$2h = 1 \text{ m}$$

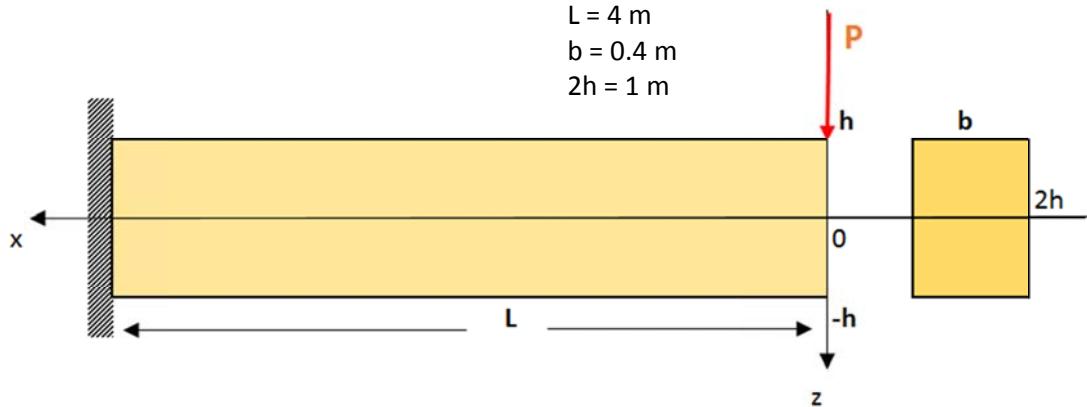


Figure 76 : exemple test 01 Poutre consol

La solution analytique des contraintes et des déformations est obtenue a partir du **principe de l'élasticité plane** qui énonce que pour un problème de contrainte plane, il est possible de trouver l'expression des contraintes, a partir d'une fonction de contrainte dite « fonction d'Airy », satisfaisant la condition d'équilibre et les conditions aux limites. [Ref101](#)

La fonction d'Airy Φ . Est donnée par
$$\Phi(x,z) = -\frac{P}{2I}xz\left(\frac{1}{3}x^2 - h^2\right)$$

avec $I = \frac{1}{12}bh^3$ d'ou $I = \frac{2}{3}bh^3$

Les expressions des contraintes

$$\sigma_{xx} = \frac{\partial^2 \Phi}{\partial x^2} ; \quad \sigma_{zz} = \frac{\partial^2 \Phi}{\partial z^2} ; \quad \sigma_{xz} = -\frac{\partial^2 \Phi}{\partial x \partial z}$$

On obtient alors

$$\sigma_{xx} = -\frac{P}{I} xz$$

$$\sigma_{zz} = 0$$

$$\sigma_{xz} = -\frac{P}{2I} (z^2 - h^2)$$

En état de déformation plane, l'expression des déformations

$$\varepsilon_{xx} = \frac{1}{E} (\sigma_{xx} - \nu \sigma_{zz}) ; \varepsilon_{zz} = \frac{1}{E} (\sigma_{zz} - \nu \sigma_{xx}) ; \varepsilon_{xz} = \frac{1+\nu}{E} \sigma_{xz}$$

$$\varepsilon_{xx} = -\frac{P}{EI} xz ; \varepsilon_{zz} = \frac{\nu P}{EI} xz ; \varepsilon_{xz} = \frac{(1+\nu)P}{2EI} (z^2 - h^2)$$

Avec l'introduction des conditions aux limites au niveau de l'encastrement (déplacement nul suivant x et z), les déplacements sont obtenus par les expressions suivantes

$$\begin{cases} u(x,z) = \int \varepsilon_{xx} dx \\ w(x,z) = \int \varepsilon_{zz} dz \end{cases}$$

Ainsi nous aurons

$$u(x,z) = -\frac{P}{2EI} [L^2 - x^2 - 2(1+\nu)h^2 + \frac{1}{3}(2+\nu)z^2]$$

$$w(x,z) = \frac{P}{2EI} [\nu xz^2 + \frac{1}{3}x^3 - L^2x + \frac{2}{3}L^3]$$

Résultats théoriques RDM	
W (y,z)	-0,002000 m
Sigma X	-6.000 Mpa

Après maillage avec **Apl_mesh_designe** avec une division de la poutre en 18 unités suivant X et 5 unités suivant Y.

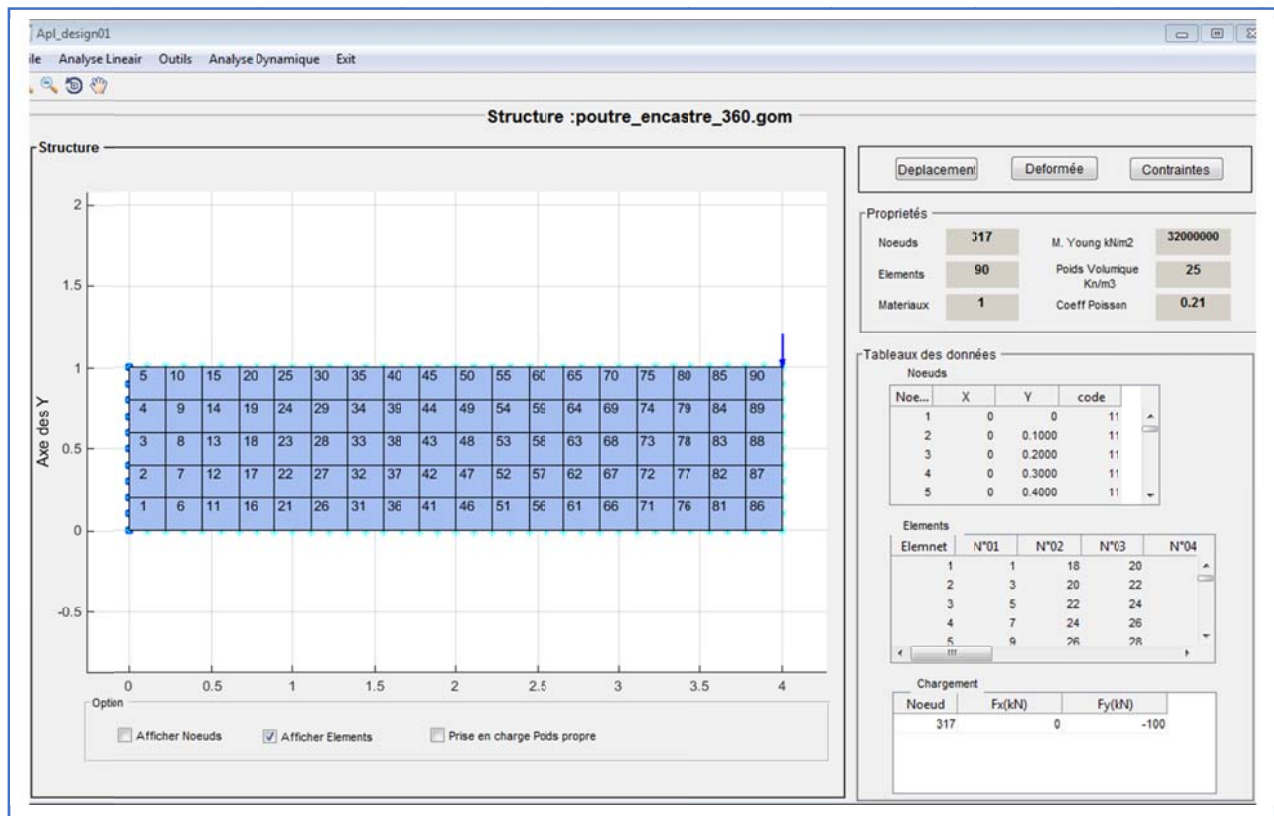


Figure 77 : maillage de la poutre avec Apl_design_Q8

Les résultats obtenus sont comme suit

Deplacement dans la structure (m)

Deplacement_Noeuds (m)		deplacement Max (m)		
	coord_X	coord_Y	deplc_X	deplc_Y
293	3.78	0.30	-1.4901063700000000e-04	-0.001914514345000
294	3.78	0.40	-7.5540861870000000e-05	-0.001914984314000
295	3.78	0.50	-2.0996033070000000e-06	-0.001916008187000
296	3.78	0.60	7.1572932940000000e-05	-0.001917166868000
297	3.78	0.70	1.4584518790000000e-04	-0.001917351037000
298	3.78	0.80	2.2198223450000000e-04	-0.001916731941000
299	3.78	0.90	3.0257782780000000e-04	-0.001918450231000
300	3.78	1.00	3.8067164130000000e-04	-0.001919585397000
301	3.89	0.00	-3.7215239720000000e-04	-0.001997591928000
302	3.89	0.20	-2.2299462580000000e-04	-0.001997785643000
303	3.89	0.40	-7.5588162870000000e-05	-0.001999314176000
304	3.89	0.60	7.1870163080000000e-05	-0.002002333524000
305	3.89	0.80	2.2136533520000000e-04	-0.002010022989000
306	3.89	1.00	3.9214690330000000e-04	-0.002009161050000
307	4.00	0.00	-3.7206898560000000e-04	-0.002080622362000
308	4.00	0.10	-2.9743165940000000e-04	-0.002080586040000
309	4.00	0.20	-2.2297074610000000e-04	-0.002080675913000
310	4.00	0.30	-1.4896302310000000e-04	-0.002080925507000
311	4.00	0.40	-7.5385447420000000e-05	-0.002081773244000
312	4.00	0.50	-1.8069366660000000e-06	-0.002083996311000
313	4.00	0.60	7.2564995210000000e-05	-0.002087285044000
314	4.00	0.70	1.4758622270000000e-04	-0.002089918591000
315	4.00	0.80	2.2224240820000000e-04	-0.002094612690000
316	4.00	0.90	3.0046890610000000e-04	-0.002113188617000
317	4.00	1.00	4.0443896430000000e-04	-0.002135123825000

Figure 78 : résultats des déplacements

- Pour le point d'extrémité de coordonnées $x = 4$ m et $y=0.5$ La valeur du déplacement obtenue par **MARS** est de
 $U_x = -0.000180693$ m
 $U_y = -0.002083996$ m
- Et pour le point d'extrémité de coordonnées $x = 0$ m et $z=0$, La valeur des contraintes obtenue par **MARS** est de
 $\sigma_x = -5.7395$ Mpa

Avec un facteur d'échelle pour l'agrandissement de 100 ; l'allure de la déformée avec **API_Design_Q8** est comme suit

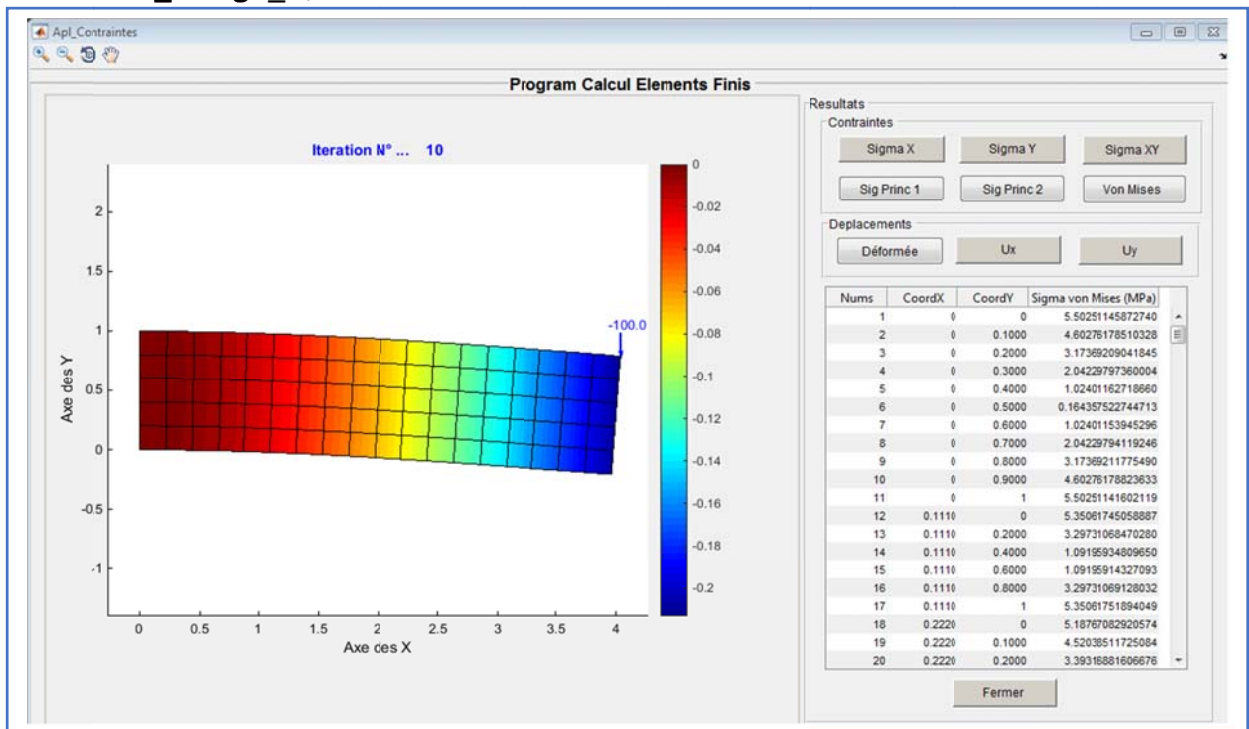


Figure 79 : Déformée et variation des contraintes par itération

Distribution des contraintes dans la poutre obtenue avec Apl design_Q8

Contraintes Sigma X

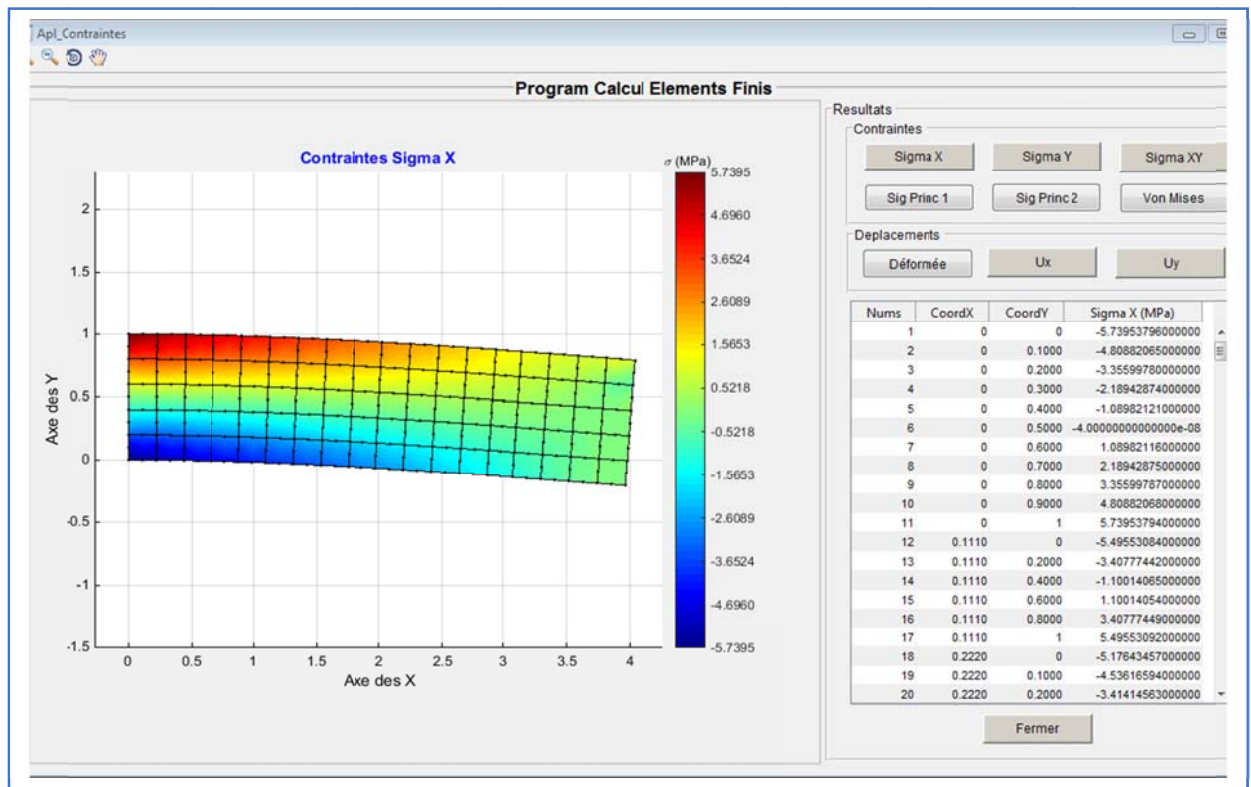


Figure 80 : Distribution des contraintes σ_x dans une poutre

Contraintes Sigma Y

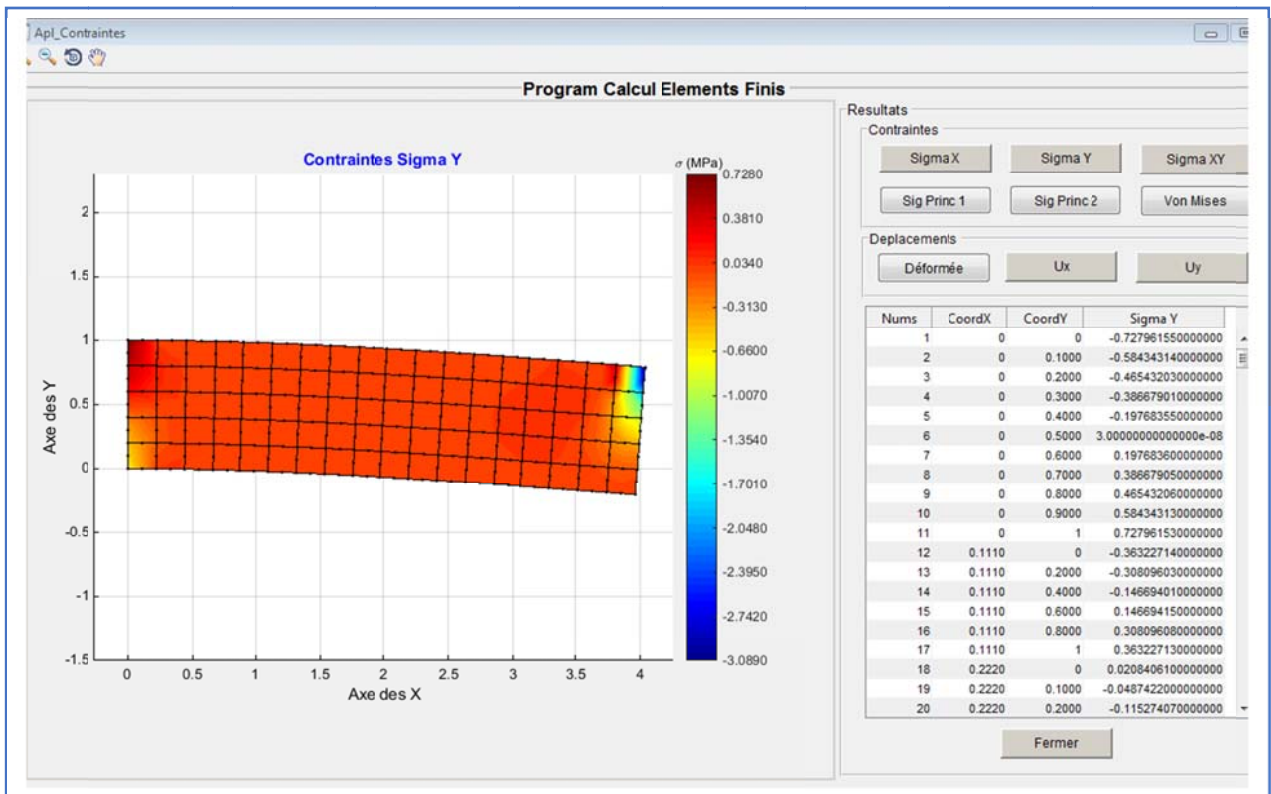


Figure 81 : Distribution des contraintes σ_y dans une poutre

Contraintes Sigma XY

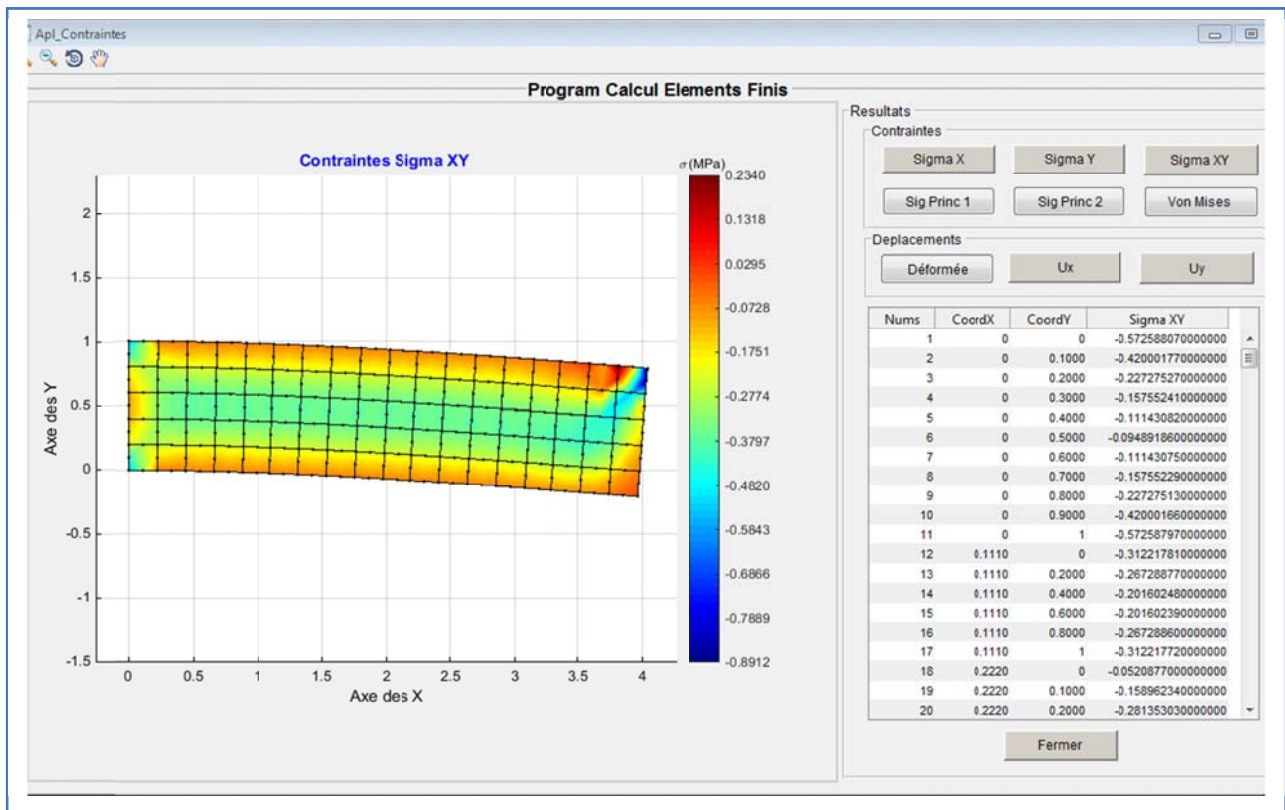


Figure 82 : Distribution des contraintes σ_{xy} dans une poutre

test 02

Pour affiner la validation de notre programme on se propose d'étudier l'exemple donné par le document de I. M. Smith, D. V. Griffiths, programming the finite element method [Ref \[11\]](#)

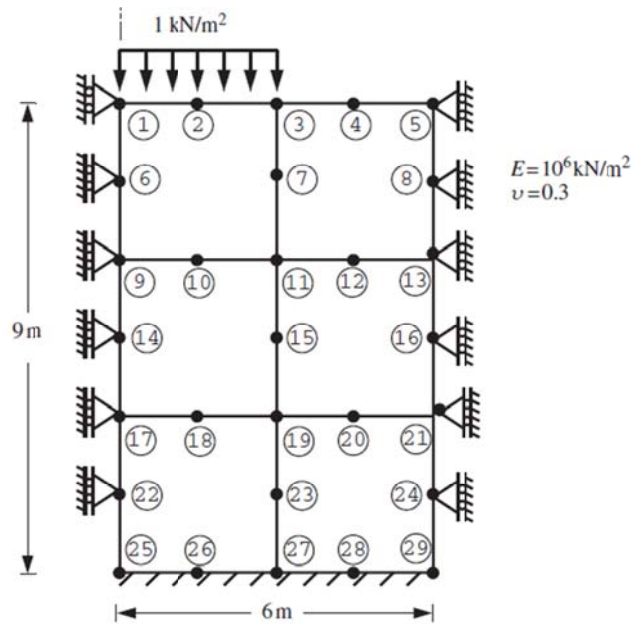


Figure 83 : exemple test02

Apl_design01

Structure :EXO515.gom

Structure

Axe des Y

Axe des X

Données Résultats Diagrammes

Propriétés

Noeuds 29 M. Young GPa 1000000

Elements 6 C.Poisson 0

Matériaux 1 P.Volum 0.3

Tableaux des données

Noeuds

Noe...	X	Y	code
1	1	1	11
2	1	2.5000	10
3	1	4	10
4	1	5.5000	10
5	1	7	10

Elements

Elemnet	N°01	N°02	N°03	N°04
1	1	12	14	
2	3	14	16	
3	5	16	18	
4	12	23	25	
5	14	25	27	

Chargement

Noeud	Fx(kN)	Fy(kN)
7	0	-5000
11	0	-20000
18	0	-5000

Afficher Elements Afficher Noeuds

Figure 84 : Affichage maillage de la structure avec conditions aux limites

Les résultats obtenus sont visualisés avec un facteur d'échelle de 10

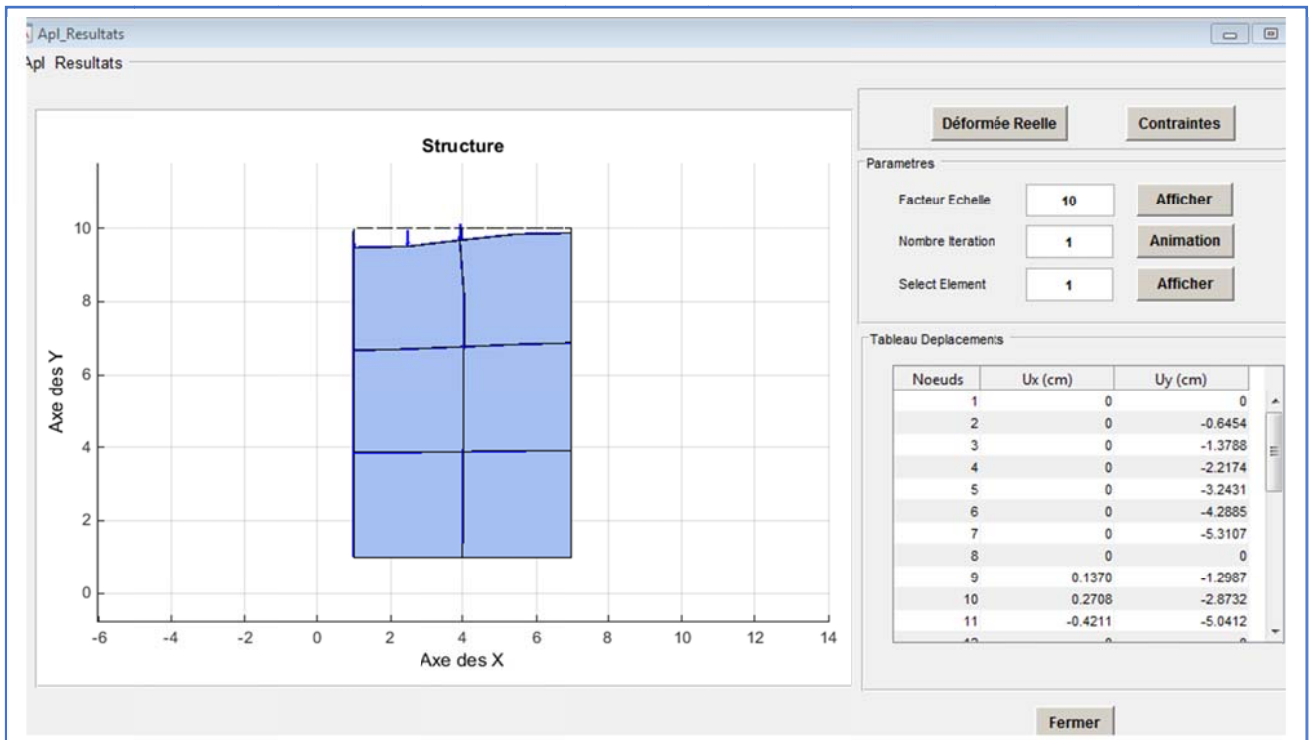


Figure 85 : Déformée de la structure avec un facteur d'échelle de 10

Les variations de contraintes obtenues σ_y

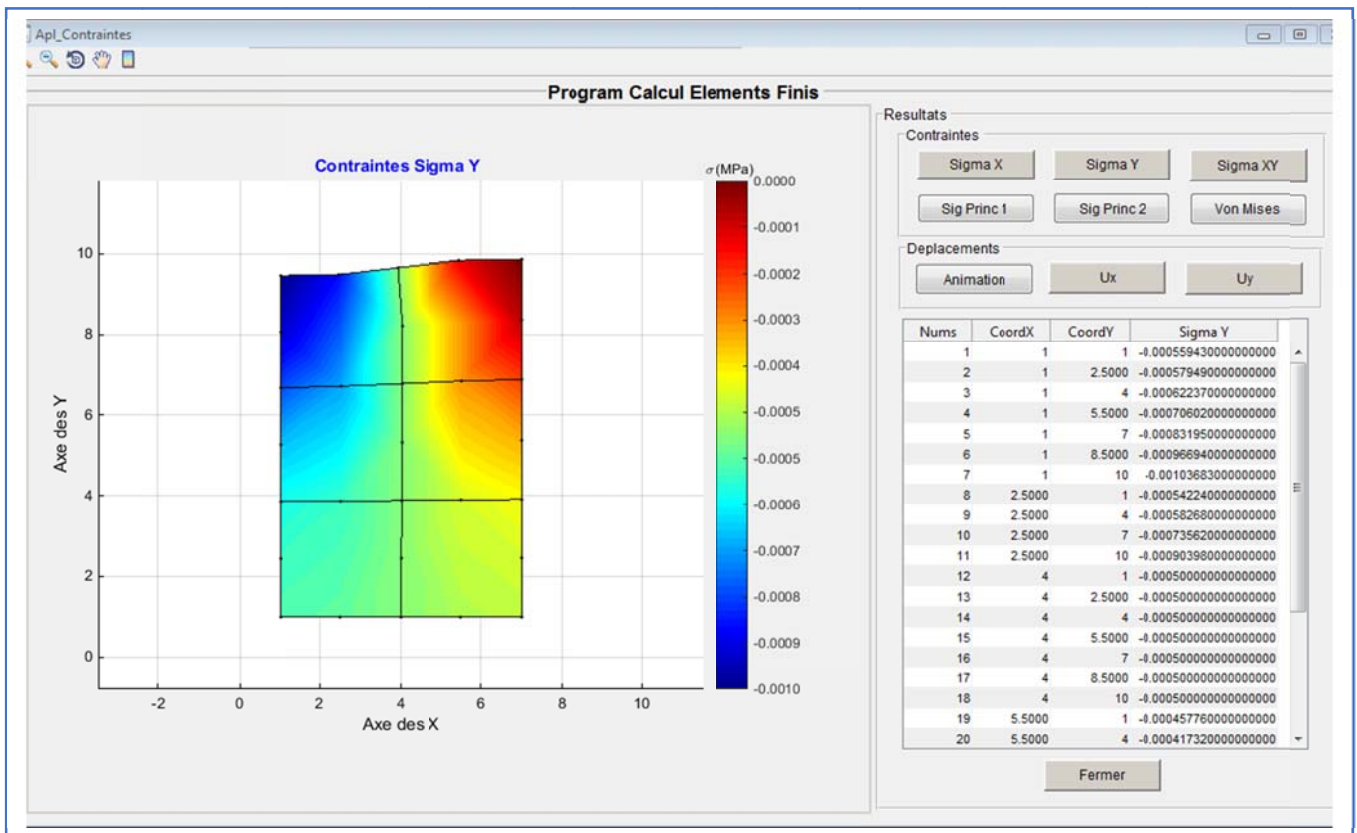


Figure 86 ; Distribution des contraintes σ_y dans une poutre

variation de la déformée suivant Uy

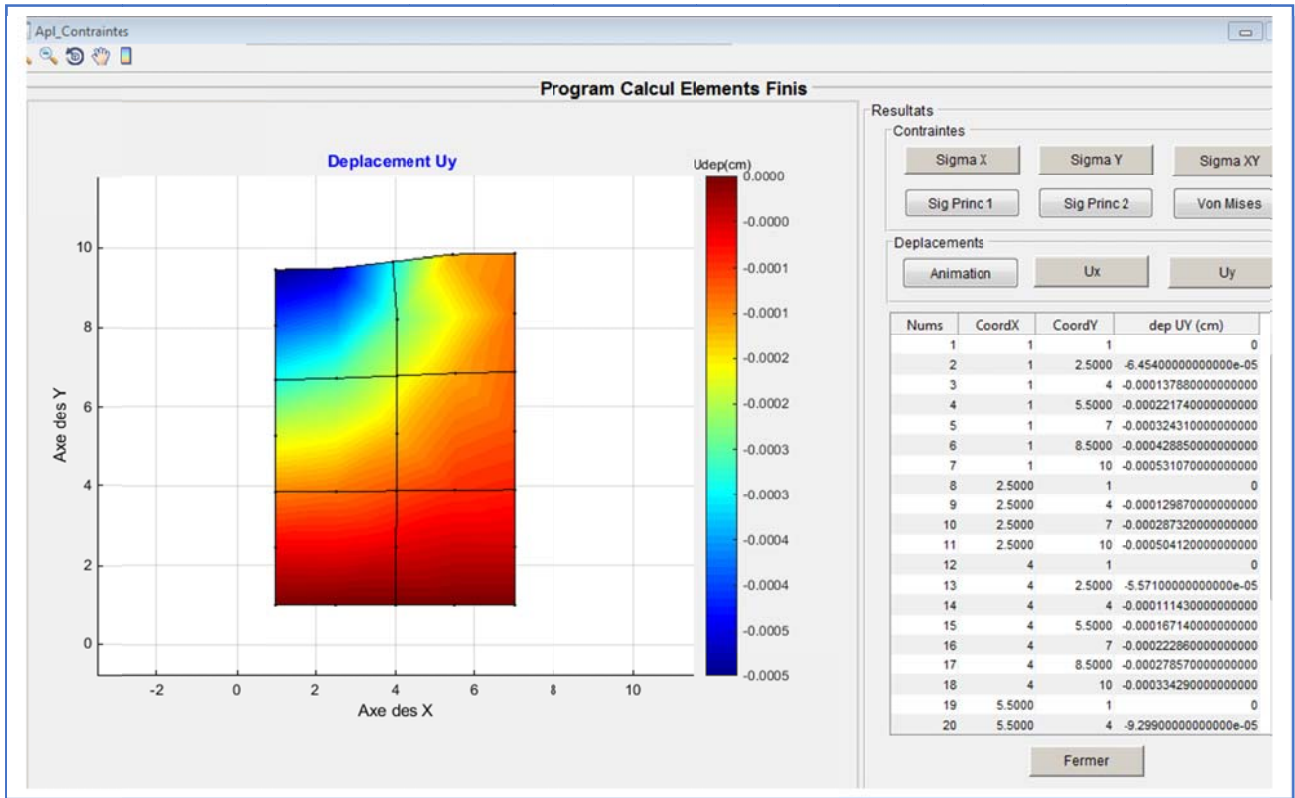


Figure 87 : variation de la déformée suivant Uy

Résultat des déplacements obtenue par Apl_design_Q8

Node:	X-Displ	Y-Displ
1.0000000000000000	-0.0000000000000000	-0.0000000000000000
2.0000000000000000	-0.0000000000000000	-0.006453529188459
3.0000000000000000	-0.0000000000000000	-0.013787735959185
4.0000000000000000	-0.0000000000000000	-0.022173783901019
5.0000000000000000	-0.0000000000000000	-0.032431230529276
6.0000000000000000	-0.0000000000000000	-0.042884837238508
7.0000000000000000	-0.0000000000000000	-0.053106720216565
8.0000000000000000	0.0000000000000000	-0.0000000000000000
9.0000000000000000	0.001370485366544	-0.012987070049713
10.0000000000000000	0.002708147711320	-0.028731660139935
11.0000000000000000	-0.004211153422117	-0.050412315382914
12.0000000000000000	0.0000000000000000	-0.0000000000000000
13.0000000000000000	0.001121994444862	-0.005571428571429
14.0000000000000000	0.001912334941441	-0.011142857142857
15.0000000000000000	0.002996313701739	-0.016714285714286
16.0000000000000000	0.003669995727547	-0.022285714285714
17.0000000000000000	0.003774202388609	-0.027857142857143
18.0000000000000000	-0.007221882550665	-0.033428571428571
19.0000000000000000	0.0000000000000000	-0.0000000000000000
20.0000000000000000	0.001370485366544	-0.009298644236001
21.0000000000000000	0.002708147711320	-0.015839768431494
22.0000000000000000	-0.004211153422117	-0.016444827474228
23.0000000000000000	0.0000000000000000	-0.0000000000000000
24.0000000000000000	0.0000000000000000	-0.004689327954398
25.0000000000000000	0.0000000000000000	-0.008497978326529
26.0000000000000000	0.0000000000000000	-0.011254787527553
27.0000000000000000	0.0000000000000000	-0.012140198042152
28.0000000000000000	0.0000000000000000	-0.012829448475778
29.0000000000000000	-0.0000000000000000	-0.013750422640578

Figure 88 : résultats des déplacements

Resultats données par l'ouvrage M. Smith, D. V. Griffiths, programming the finite element method Ref [1]

4.5 Apl design T6

Soit à étudier la même poutre consol étudiée pour l'élément Q8 encadrée à l'une de ses extrémités et libre de l'autre, soumise à une charge concentrée $P=100$ kN, de longueur $L = 4.00$ m, de section rectangulaire $b = 0.4$ m, $h = 0.5$ m, et sans tenir compte du poids propre de la structure

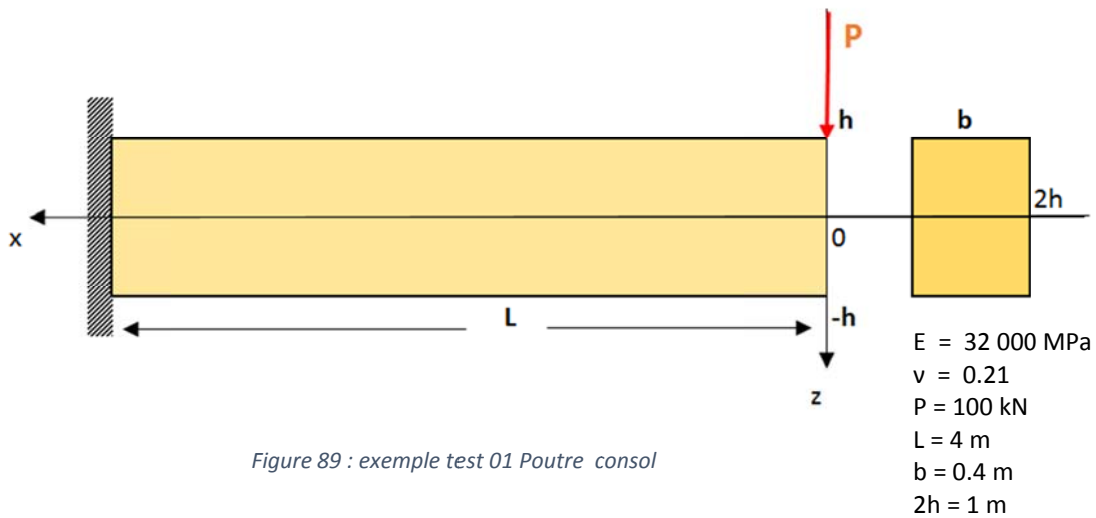


Figure 89 : exemple test 01 Poutre consol

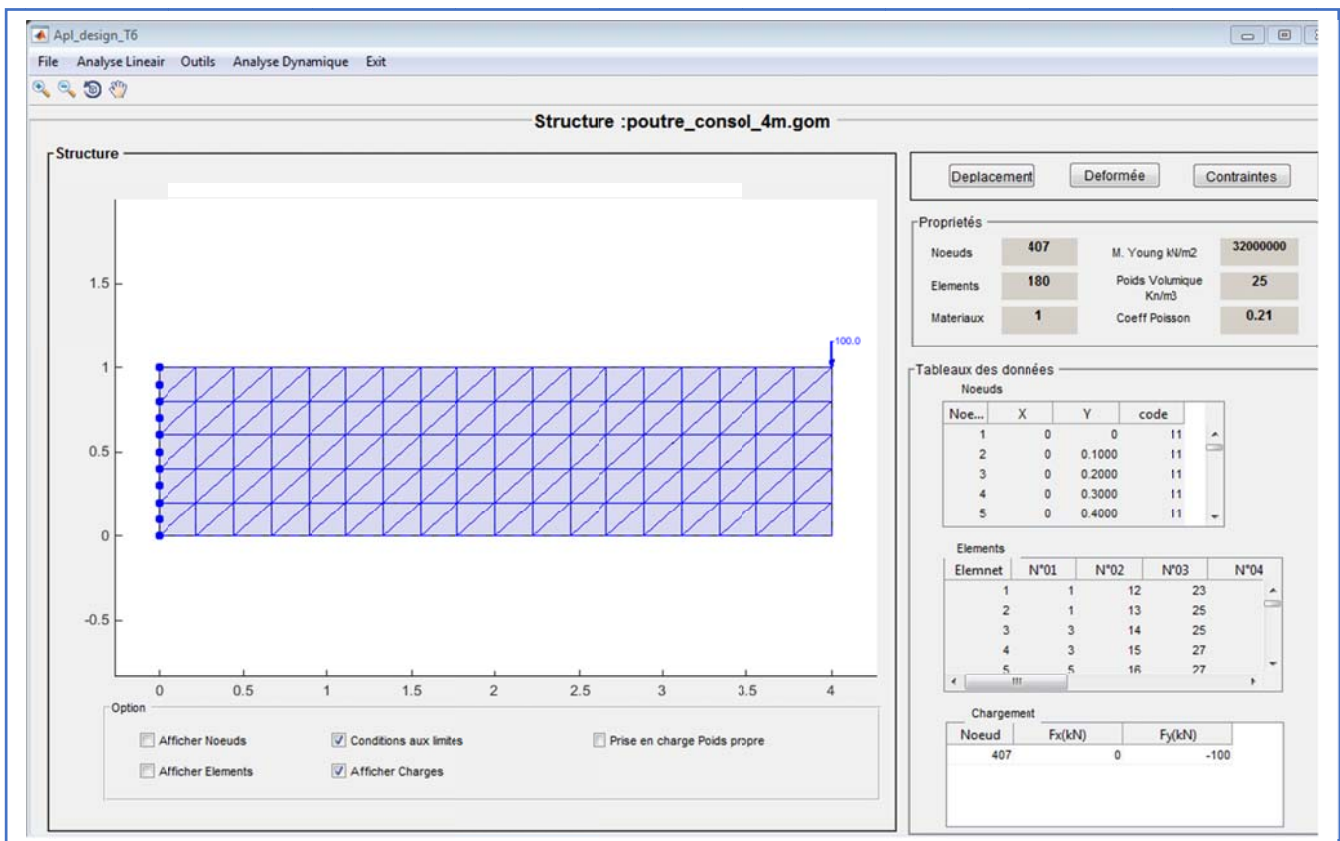


Figure 90 : maillage de la poutre avec Apl_design T6

Les déplacements

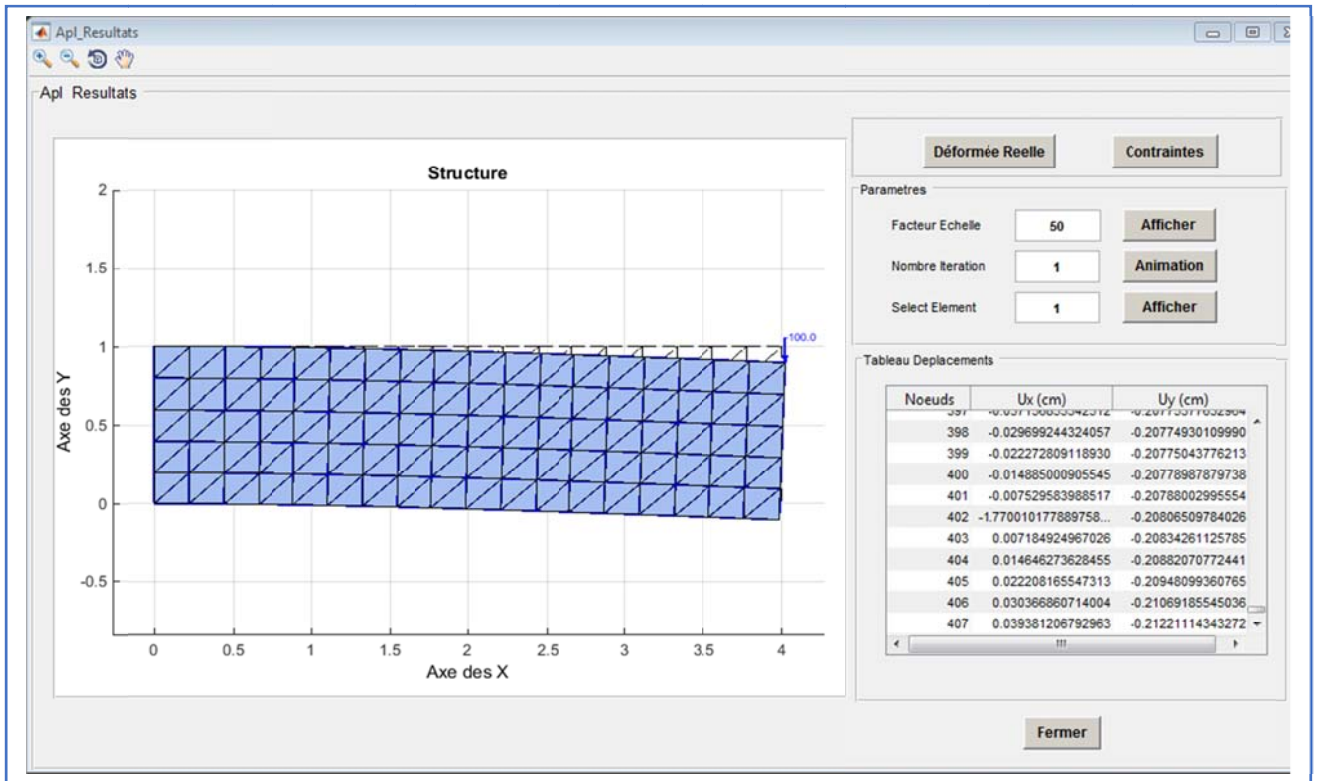


Figure 91 : Déformée de la structure

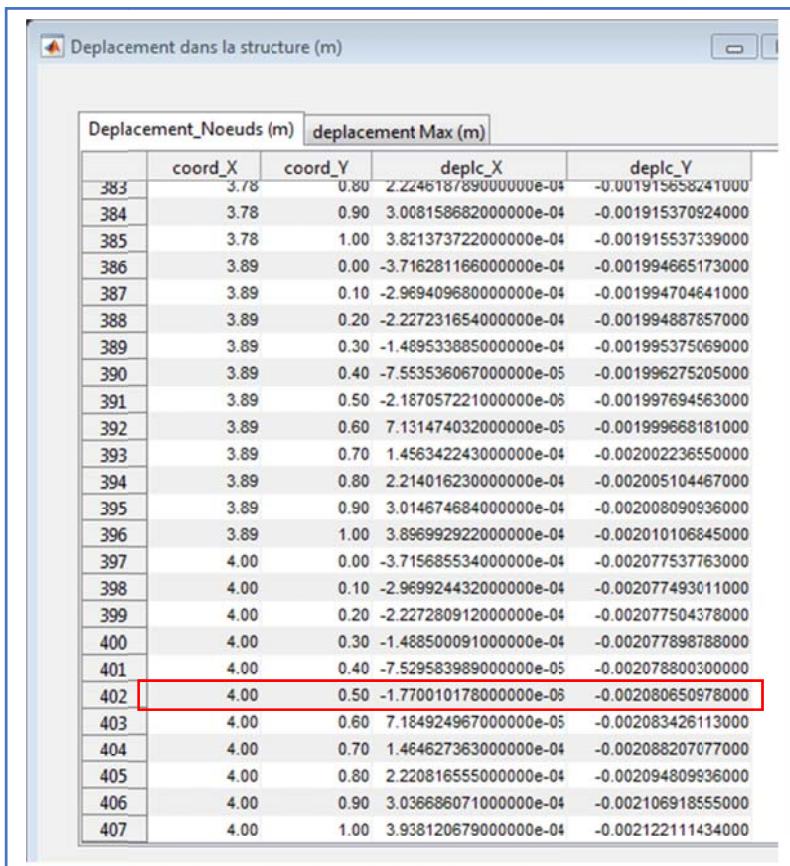


Figure 92 : résultats des déplacements

- Pour le point d'extrémité de coordonnées x = 4 m et y=0.5 La valeur du déplacement obtenue par **MARS** est de
 $U_x = -0.0001770$ m
 $U_y = -0.0020806$ m
- Et pour le point d'extrémité de coordonnées x = 0 m et z=0 , La valeur des contraintes obtenue par **MARS** est de
 $\sigma_x = -5.55317$ Mpa

Les déplacements

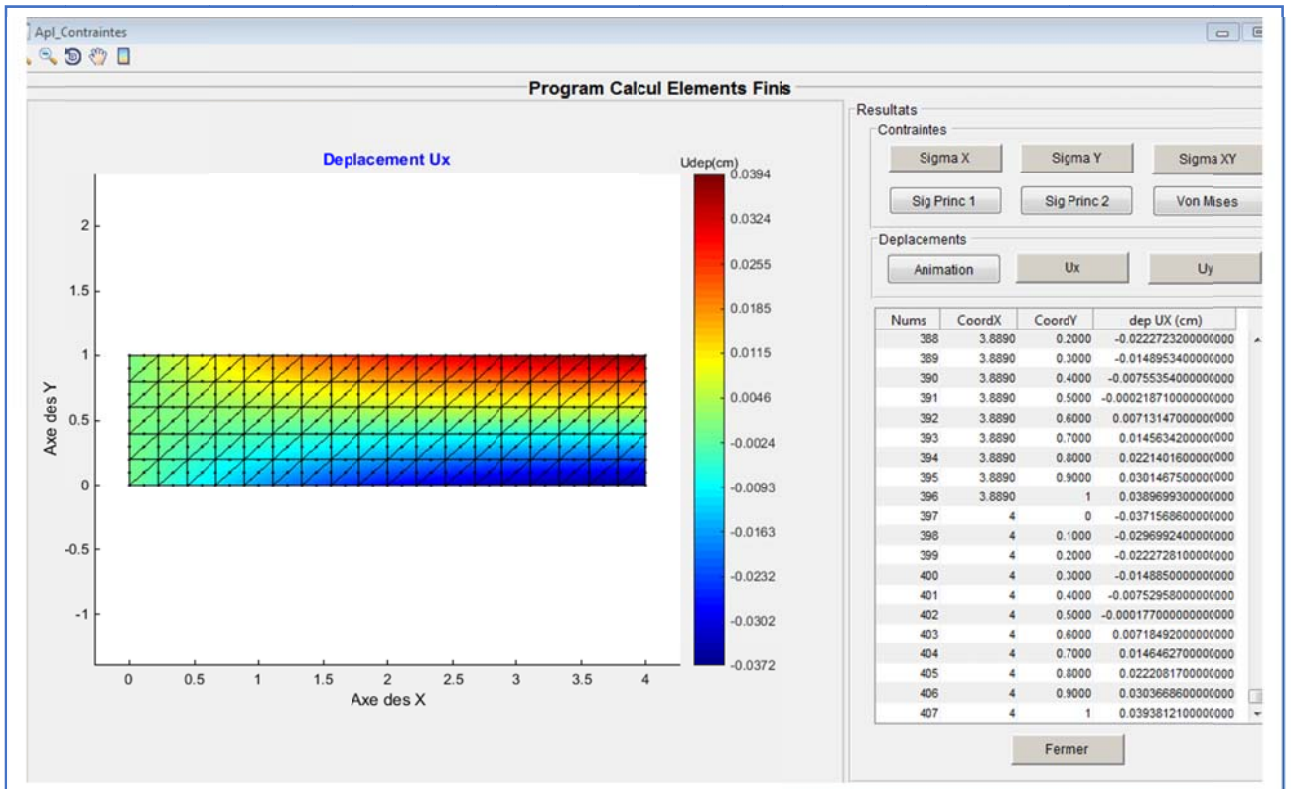


Figure 93 : Déplacement et variation des contraintes

Les contraintes

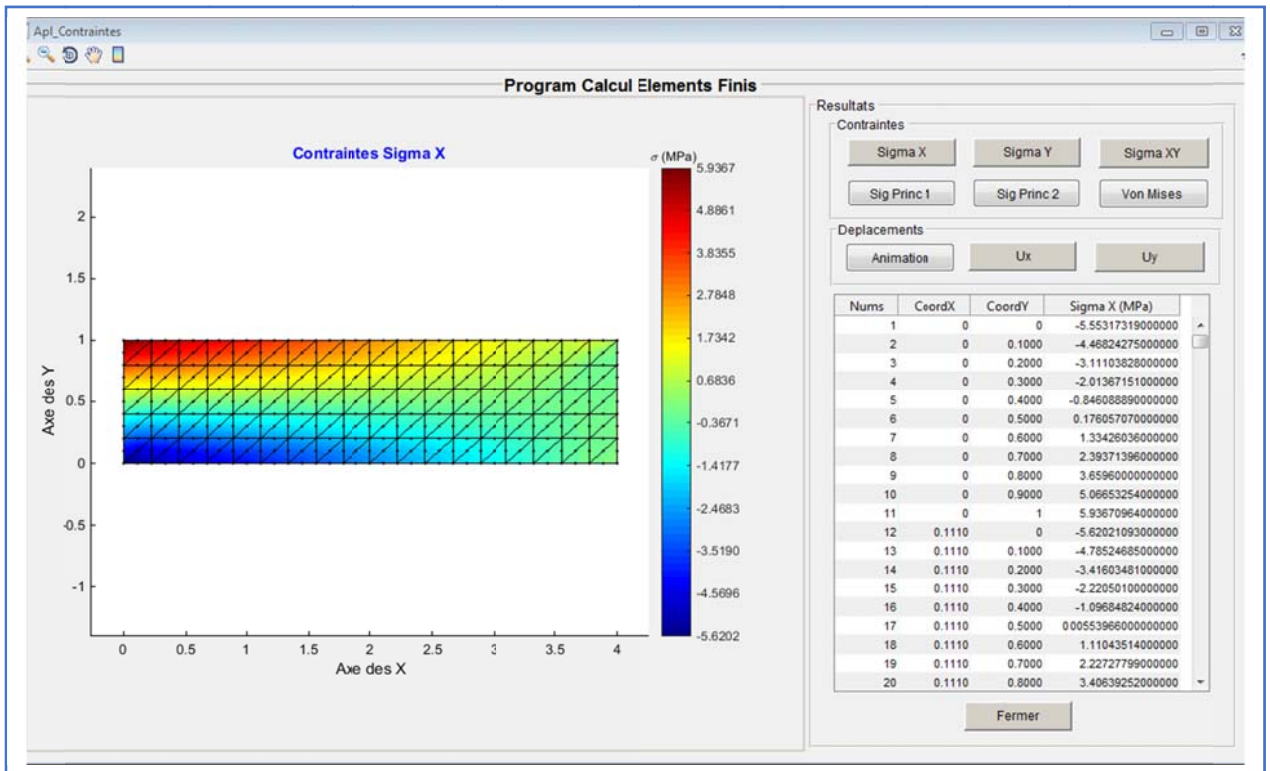


Figure 94 : Distribution des contraintes σ_x

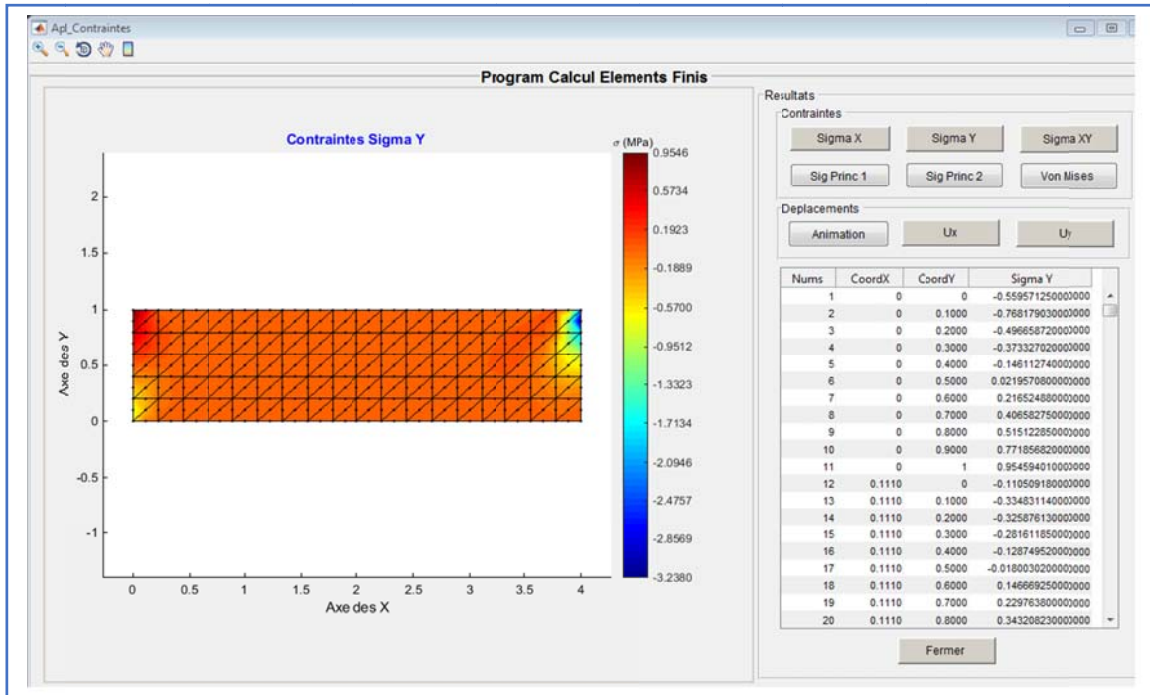


Figure 95 : Distribution des contraintes σ_y

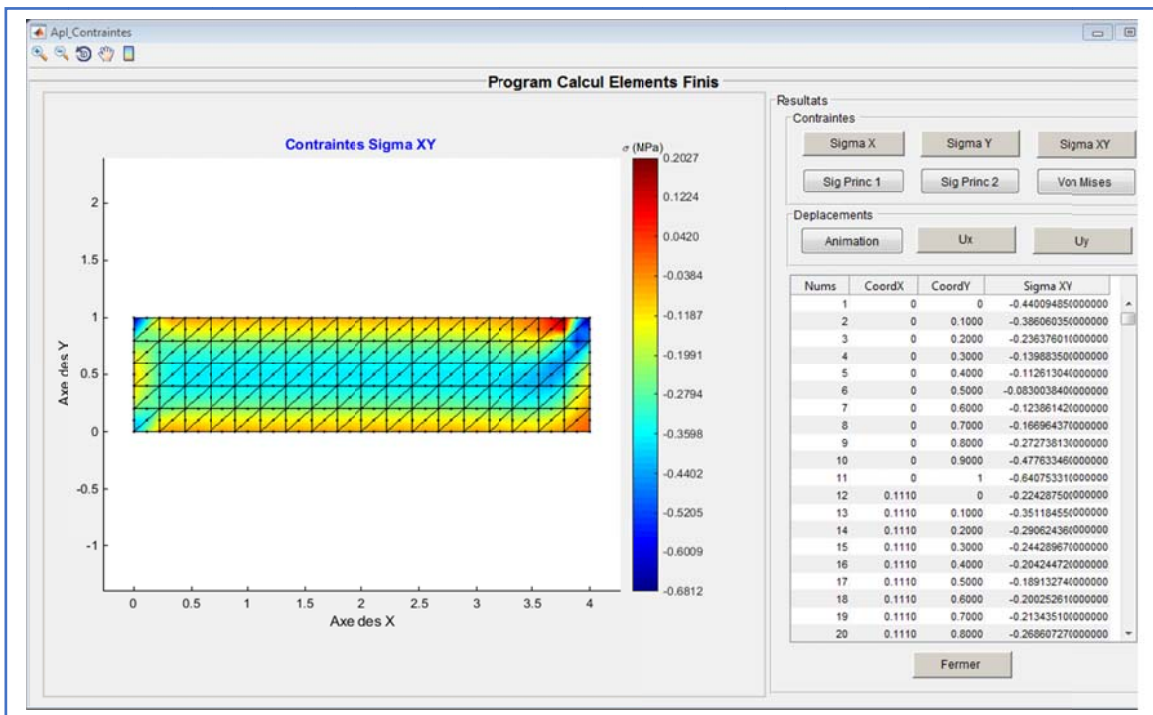


Figure 96 : Distribution des contraintes σ_{xy}

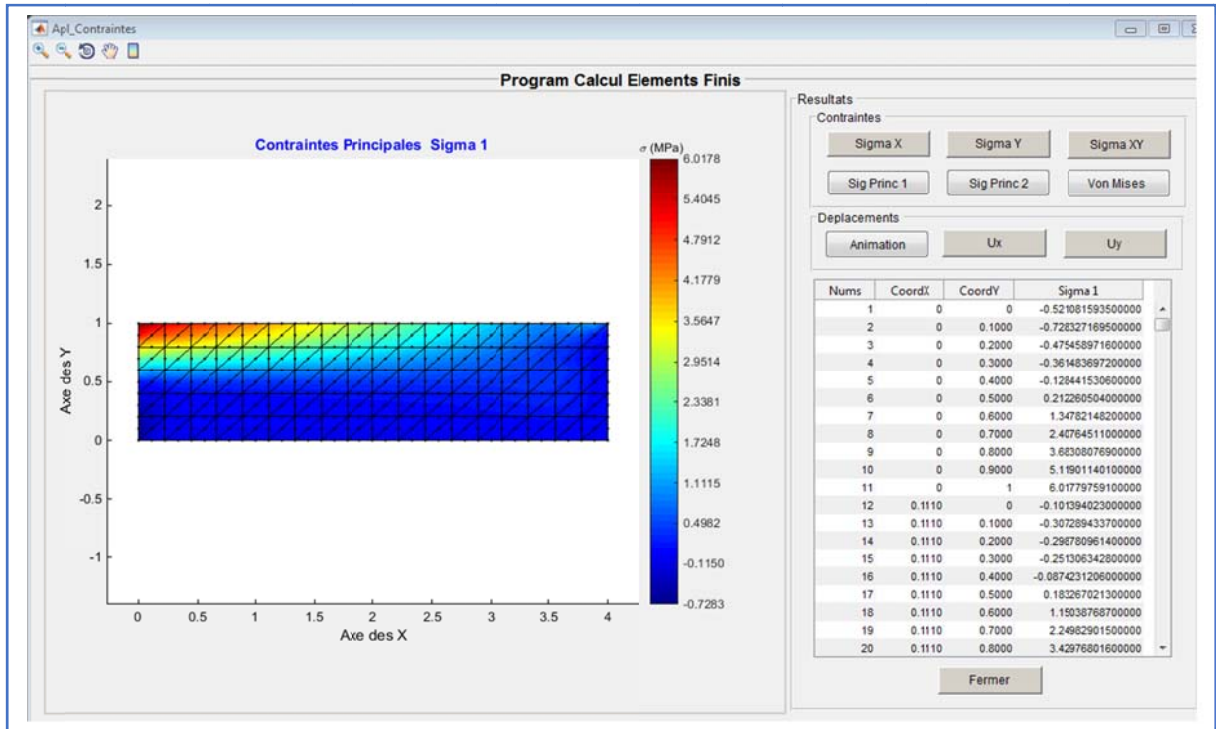


Figure 97 : Distribution des contraintes σ principale 1

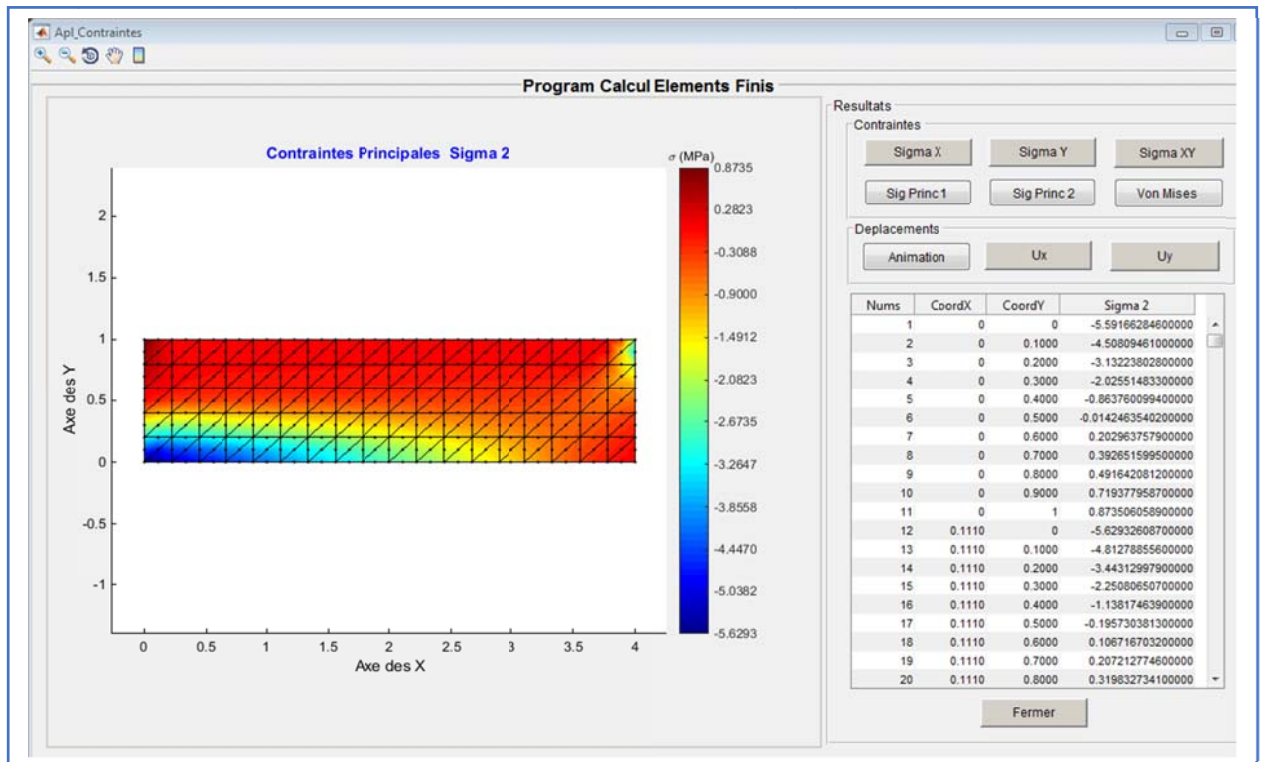


Figure 98 : Distribution des contraintes σ principale 2

Contraintes au nœuds de la structure

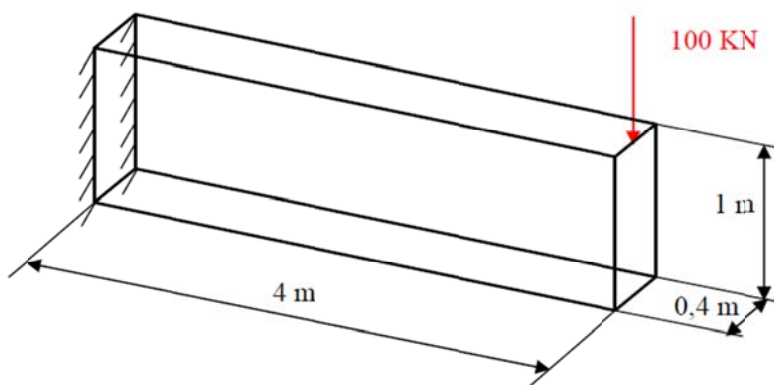
	X_Strs	Y_Strs	XY_Strs
1	-5.553173191000000	-0.559571249000000	-0.440094854300000
2	-4.468242748000000	-0.768179030400000	-0.386060346700000
3	-3.111038277000000	-0.496658718000000	-0.236376007200000
4	-2.013671509000000	-0.373327016100000	-0.139883504600000
5	-0.846088887200000	-0.146112738700000	-0.112613037400000
6	0.176057065100000	0.021957075740000	-0.083003837360000
7	1.334260359000000	0.216524876700000	-0.123861423900000
8	2.393713965000000	0.406532748200000	-0.166964367800000
9	3.659600000000000	0.515122845200000	-0.272738128300000
10	5.066532538000001	0.771856816800000	-0.477633463200000
11	5.936709642000000	0.954594008500000	-0.640753307000000
12	-5.620210932000000	-0.110509177500000	-0.224287503700000
13	-4.785246855000001	-0.334831139900000	-0.351184554000000
14	-3.416034813000000	-0.325876131500000	-0.290624357500000
15	-2.220500995000000	-0.281611845000000	-0.244289668800000
16	-1.096848240000000	-0.128749524200000	-0.204244724900000
17	0.005539660022000	-0.018003024140000	-0.189132742600000
18	1.110435143000000	0.146689247100000	-0.200252612100000
19	2.227277990000000	0.229783798100000	-0.213435104200000
20	3.406392519000000	0.343208225200000	-0.268607270600000
21	4.618019719000000	0.276454203700000	-0.229042893400000
22	5.672801211000000	0.461000713700000	-0.379284794800000
23	-5.095110942000001	-0.003487871450000	-0.056667200380000
24	-4.527921556000000	-0.043592213300000	-0.148003588000000
25	-3.410954844000000	-0.096435111690000	-0.276784556300000

Figure 99 : Résultats des déplacements

4.6 Apl design Tetra4

Test 01

On se propose d'étudier la flexion d'une poutre encastree par une de ses extremités et libre de l'autre, soumise à une charge concentree $P=100$ KN, de longueur $L = 4.00$ m, de section rectangulaire de largeur 0.4 m et hauteur de 1 m, et cela sans tenir compte du poids propre de la structure



$E = 32\,000$ MPa
 $\nu = 0.21$
 $P = 100$ kN
 $L = 4$ m
 $b = 0.4$ m
 $h = 1$ m

Figure 100 : exemple test 01 Poutre consol

Après maillage avec l'élément tétraèdre T4

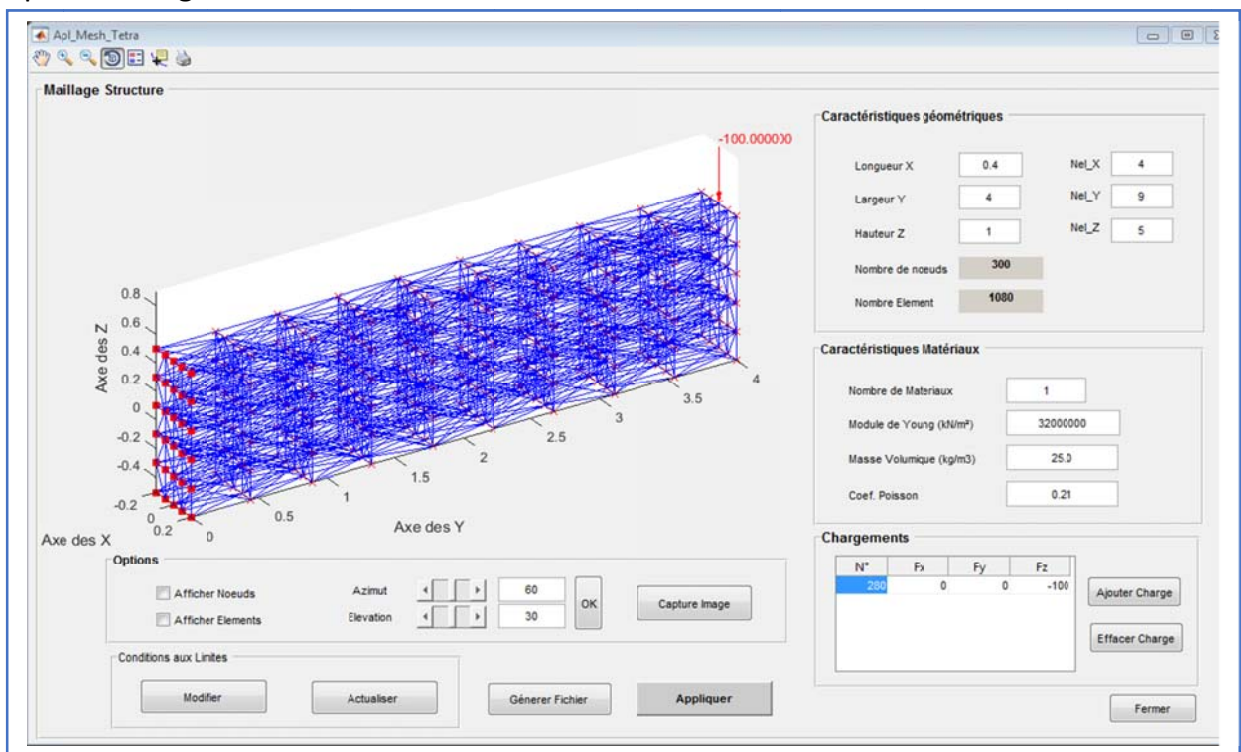


Figure 101 : maillage de la structure

Les résultats obtenus déformée et déplacements suivant x et y

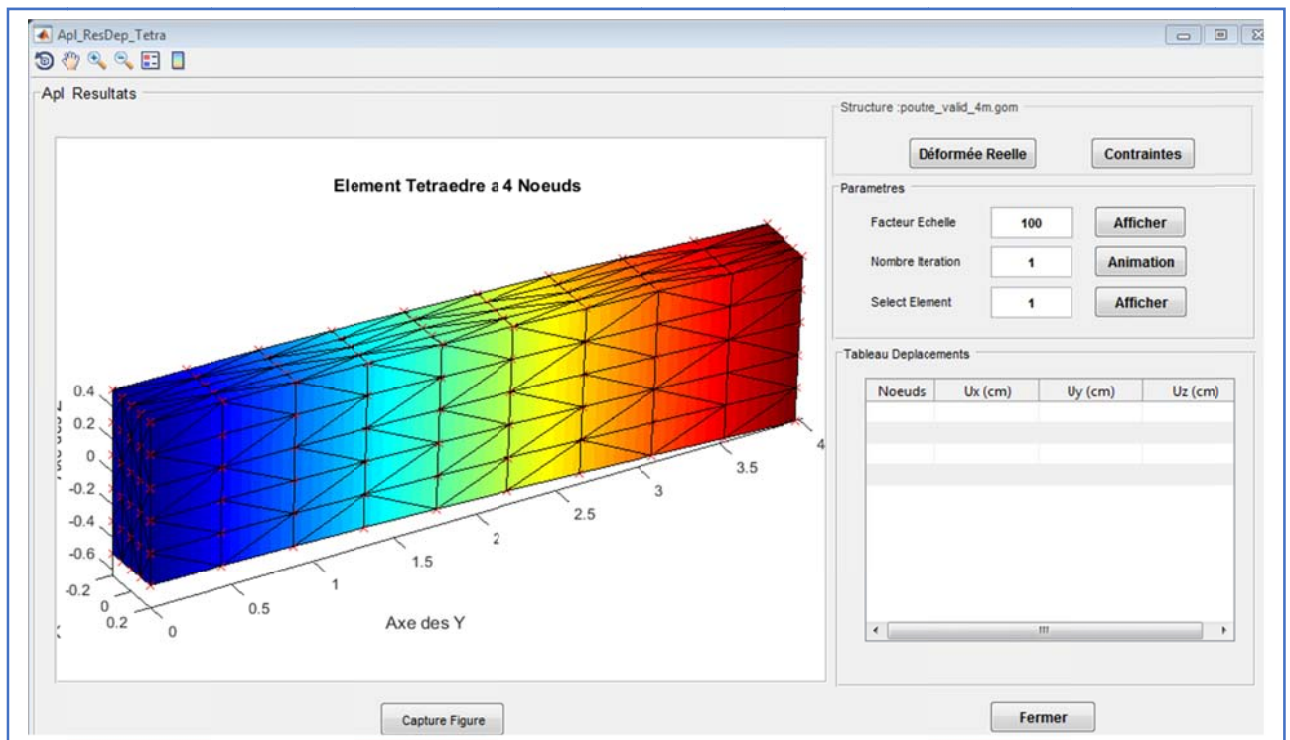


Figure 102 :Déformée de la structure

deplacement aux noeuds de la structure

Deplacement_Noeuds (cm)		deplacement Max (cm)				
	coord_X	coord_Y	coord_Z	deplc_X	deplc_Y	deplc_Z
251	-0.2000	0.5000	0.5000	0.0000	0.0000	0.0000
252	-0.2000	0.4444	0.5000	1.7414e-04	0.0052	-0.0036
253	-0.2000	0.8889	0.5000	2.7210e-04	0.0099	-0.0108
254	-0.2000	1.3333	0.5000	4.4423e-04	0.0137	-0.0224
255	-0.2000	1.7778	0.5000	7.4140e-04	0.0175	-0.0367
256	-0.2000	2.2222	0.5000	0.0010	0.0198	-0.0539
257	-0.2000	2.6667	0.5000	0.0010	0.0218	-0.0732
258	-0.2000	3.1111	0.5000	0.0012	0.0233	-0.0941
259	-0.2000	3.5556	0.5000	0.0014	0.0244	-0.1162
260	-0.2000	4	0.5000	0.0015	0.0249	-0.1392
261	-0.1000	0	0.5000	0	0	0
262	-0.1000	0.4444	0.5000	-4.7848e-05	0.0049	-0.0034
263	-0.1000	0.8889	0.5000	9.3586e-05	0.0101	-0.0109
264	-0.1000	1.3333	0.5000	2.8663e-04	0.0134	-0.0223
265	-0.1000	1.7778	0.5000	6.2869e-04	0.0173	-0.0367
266	-0.1000	2.2222	0.5000	8.6656e-04	0.0199	-0.0540
267	-0.1000	2.6667	0.5000	9.7639e-04	0.0218	-0.0733
268	-0.1000	3.1111	0.5000	0.0012	0.0234	-0.0941
269	-0.1000	3.5556	0.5000	0.0013	0.0243	-0.1162
270	-0.1000	4	0.5000	0.0014	0.0248	-0.1394
271	0	0	0.5000	0	0	0
272	0	0.4444	0.5000	-2.1348e-04	0.0051	-0.0034
273	0	0.8889	0.5000	-8.6670e-05	0.0099	-0.0110
274	0	1.3333	0.5000	1.3497e-04	0.0137	-0.0223
275	0	1.7778	0.5000	5.1158e-04	0.0173	-0.0367

Figure 103 : résultats des déplacement

- Pour le point d'extrémité de coordonnées $x = -0.2$ et $y=4$ $z=0.5$ La valeur du déplacement obtenue par **MARS** est de
 $U_x = -0.0015$ cm
 $U_y = -0.0249$ cm
 $U_z = -0.1392$ cm
- Et pour le point d'extrémité de coordonnées $x = -0.2$ m et $y=0$ $z=-0.5$, La valeur des contraintes obtenue par **MARS** est de
 $\sigma_x = -3.9285$ Mpa

Contraintes

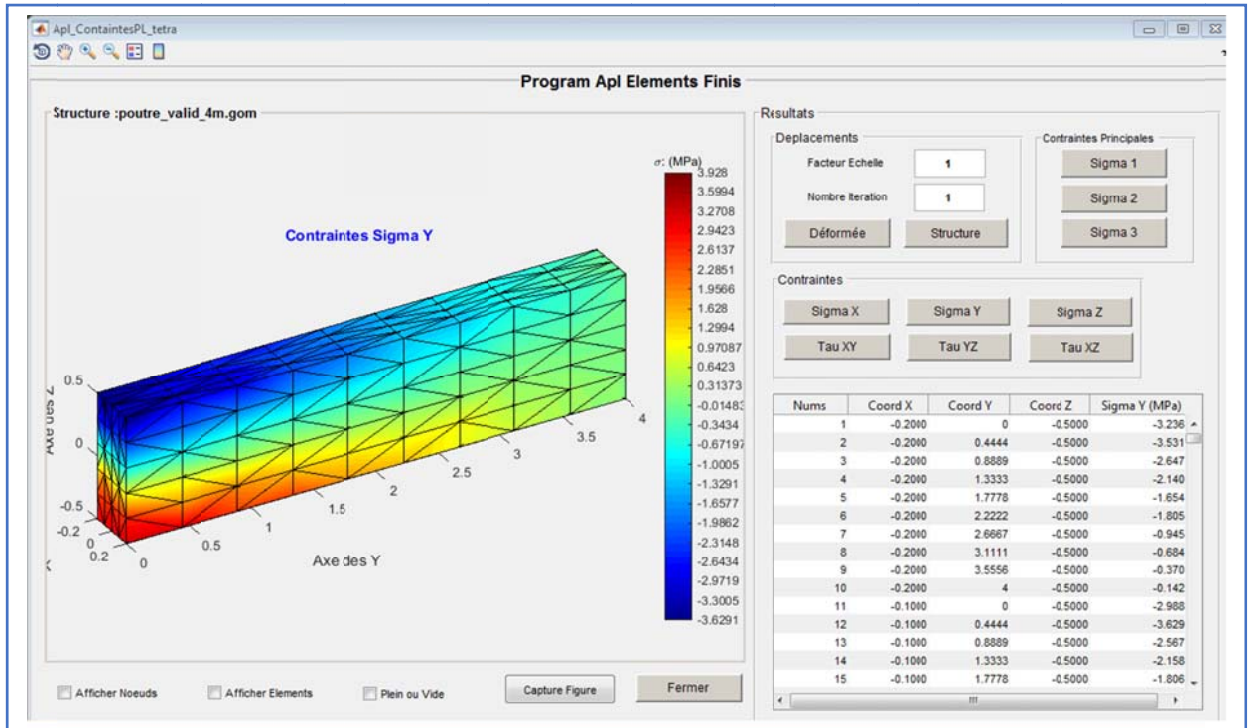


Figure 104 : distribution des contraintes σ_y

Contraintes dans la structure

	Contrainte_Noeuds (MPa)		Contrainte_PGauss (MPa)		Contrainte_centre_PG (MPa)	
	Sig_X	Sig_Y	Sig_Z	Tau_XY	Tau_YZ	Tau_XZ
37	-0.0703	-1.2120	-0.0554	-0.0281	0.0214	-0.0145
38	0.0242	-0.5915	0.0182	0.0261	-0.1160	0.0024
39	-0.0132	-0.3922	0.0038	-0.0154	-0.1041	8.1143e-04
40	0.0300	-0.1379	5.3864e-04	0.0053	-0.1339	-0.0031
41	-0.3783	-3.3232	-0.4650	-0.2153	-0.3458	-0.0557
42	-0.0437	-3.1683	-0.2674	-0.0406	-0.1507	-0.0094
43	0.0181	-2.5290	0.1016	-0.0233	-0.0755	-0.0261
44	-0.1325	-2.4835	0.0160	-0.3210	-0.3183	-0.0850
45	-0.0644	-1.8138	-0.0854	-0.0810	-0.2079	0.0072
46	-0.0726	-1.5325	-0.0708	-0.0223	0.0756	-0.0239
47	-0.0173	-0.9068	0.0390	-0.2012	-0.1292	0.0200
48	-0.0097	-0.6813	-0.0022	-0.0579	-0.1205	0.0126
49	0.0091	-0.3584	0.0016	0.0225	-0.0985	0.0024
50	0.0358	-0.1399	-0.0088	0.0553	-0.1255	-0.0364
51	-0.5070	-2.5398	-0.6398	-0.0728	-0.8568	-0.0115
52	-0.0214	-2.2794	-0.1182	-0.0789	-0.2318	0.0226
53	-0.0341	-1.6962	-0.0853	0.0544	0.0077	-0.0212
54	-0.0529	-1.5677	-0.0286	-0.0315	0.0420	0.0249
55	-0.0045	-0.8710	0.0045	0.0338	-0.3111	0.0216
56	-0.0303	-1.1483	-0.0239	-0.0363	-0.1865	0.0104
57	-0.0016	-0.8302	-0.0403	0.0022	-0.0678	0.0033
58	-0.0013	-0.4153	0.0259	0.0246	-0.2770	-0.0037
59	0.0149	-0.2435	-0.0347	-0.0090	-0.1941	-0.0048

Figure 105 : résultat des contraintes

Test 02

Soit l'exemple donné dans l'ouvrage I. M. Smith, D. V. Griffiths, L. Margetts - programming the finite element method [Ref \[1\]](#)

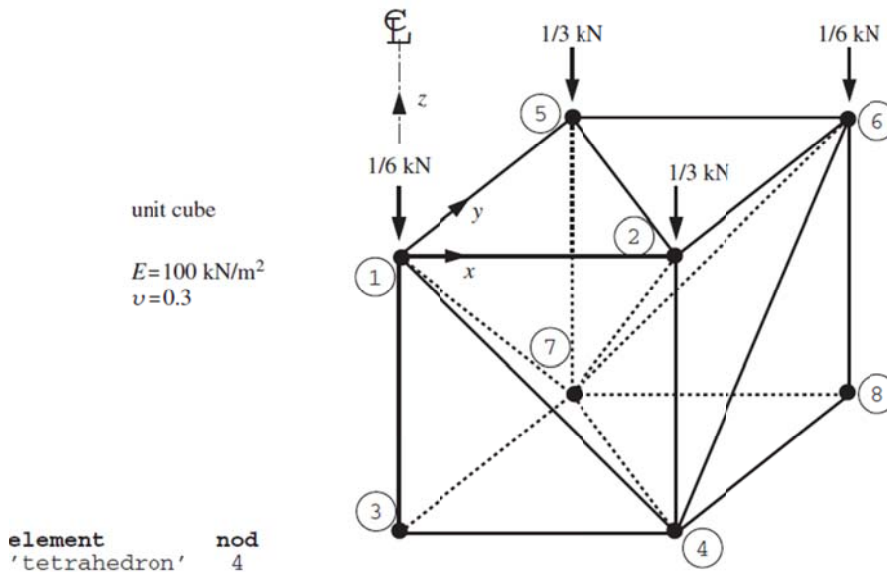


Figure 106 : exemple test 02 cube tétraèdre

I. M. Smith, D. V. Griffiths, L. Margetts - programming the finite element method

Après maillage est préparation du fichier de données, la structure est affichée dans l'interface graphique de notre programme .

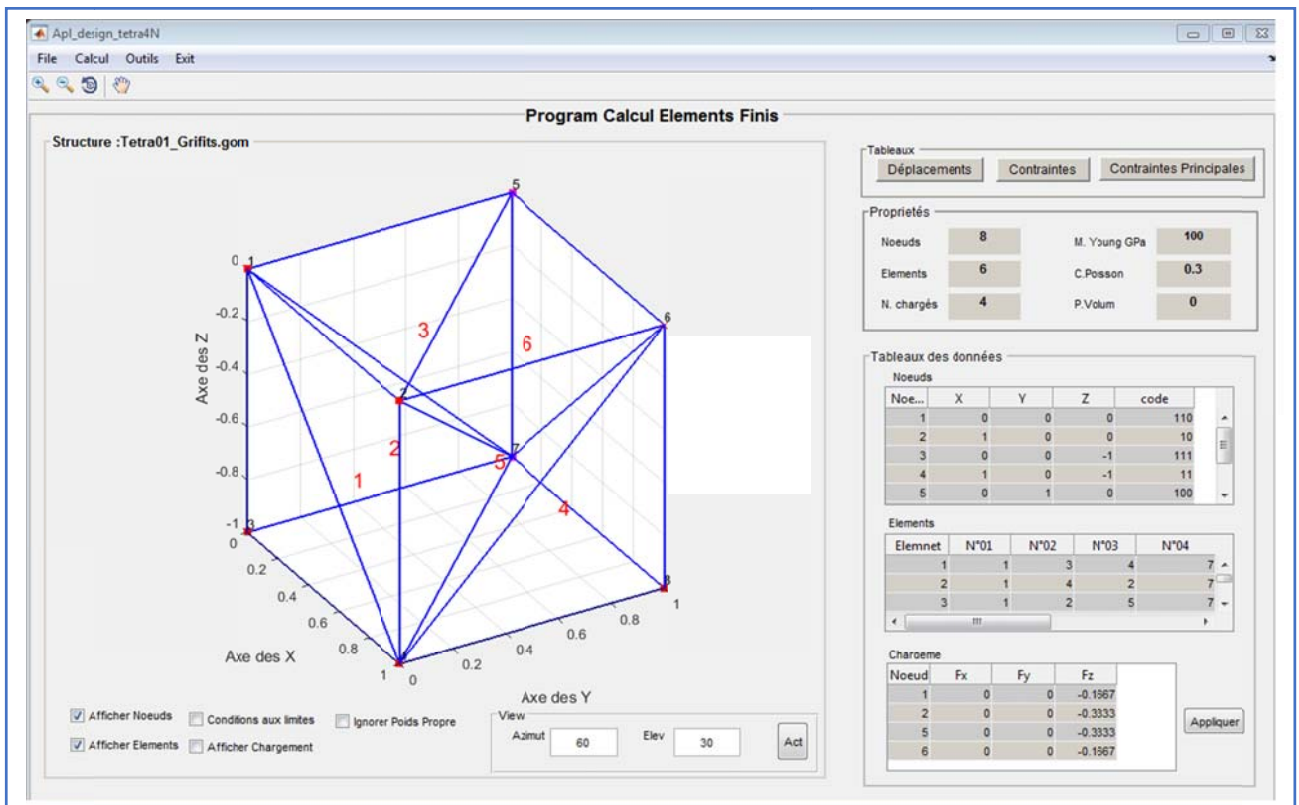


Figure 107 : maillage de cube tetraedre

Après exécution du programme on obtient les résultats suivants

	coord_X	coord_Y	coord_Z	deplc_X	deplc_Y	deplc_Z
1	0	0	0	0	0	0.0100
2	1	0	0	-0.0030	0	0.0100
3	0	0	-1	0	0	0
4	1	0	-1	-0.0030	0	0
5	0	1	0	0	-0.0030	0.0100
6	1	1	0	-0.0030	-0.0030	0.0100
7	0	1	-1	0	-0.0030	0
8	1	1	-1	-0.0030	-0.0030	0

Figure 108 : Résultats obtenus

	Sig_Xgp	Sig_Ygp	Sig_Zgp	Tau_XYgp	Tau_YZgp	Tau_XZgp
1	1.9645e-05	2.1005e-05	1.0001	0	0	0
2	-2.3022e-06	-1.3892e-05	1.0000	6.4745e-06	-2.9739e-05	-2.3264e-05
3	-1.2305e-05	-2.0745e-06	0.9999	0	-3.6407e-05	-2.9739e-05
4	1.0873e-05	9.0227e-06	1.0001	-1.5562e-06	7.4665e-06	9.3164e-06
5	-6.1024e-06	1.2970e-06	1.0000	8.7522e-06	2.5414e-05	3.1889e-05
6	-9.8087e-06	-1.5358e-05	0.9999	-2.1576e-06	3.6324e-05	4.2992e-05

les résultats du calcul donné par Griffiths. [Ref \[1\]](#)

Node	x-disp	y-disp	z-disp
1	0.0000E+00	0.0000E+00	-0.1000E-01
2	0.3000E-02	0.0000E+00	-0.1000E-01
3	0.0000E+00	0.0000E+00	0.0000E+00
4	0.3000E-02	0.0000E+00	0.0000E+00
5	0.0000E+00	0.3000E-02	-0.9999E-02
6	0.3000E-02	0.3000E-02	-0.1000E-01
7	0.0000E+00	0.3000E-02	0.0000E+00
8	0.3000E-02	0.3000E-02	0.0000E+00

```
The integration point (nip= 1) stresses are:
```

Element	x-coord	y-coord	z-coord	tau_yz	tau_zx
sig_x	sig_y	sig_z	tau_xy		
1	0.2500E+00	0.2500E+00	-0.7500E+00		
-0.1965E-04	-0.2100E-04	-0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00
2	0.5000E+00	0.2500E+00	-0.5000E+00		
0.2302E-05	0.1389E-04	-0.1000E+01	-0.6475E-05	0.2974E-04	0.2326E-04
3	0.2500E+00	0.5000E+00	-0.2500E+00		
0.1230E-04	0.2075E-05	-0.9999E+00	0.0000E+00	0.3641E-04	0.2974E-04
4	0.7500E+00	0.7500E+00	-0.7500E+00		
-0.1087E-04	-0.9023E-05	-0.1000E+01	0.1556E-05	-0.7467E-05	-0.9316E-05
5	0.7500E+00	0.5000E+00	-0.5000E+00		
0.6102E-05	-0.1297E-05	-0.1000E+01	-0.8752E-05	-0.2541E-04	-0.3189E-04
6	0.5000E+00	0.7500E+00	-0.2500E+00		
0.9809E-05	0.1536E-04	-0.9999E+00	0.2158E-05	-0.3632E-04	-0.4299E-04

Figure 109 : résultats données par Griffiths

4.7 Apl design Hexa20

Test 01

On se propose d'étudier la flexion d'une poutre encastree a l'une de ses extremités et libre de l'autre, soumise a une charge concentree $P=100$ KN, de longueur $L = 4.00$ m, de section rectangulaire de largeur 0.4 m, avec une hauteur de 1 m, et sans tenir compte du poids propre de la structure

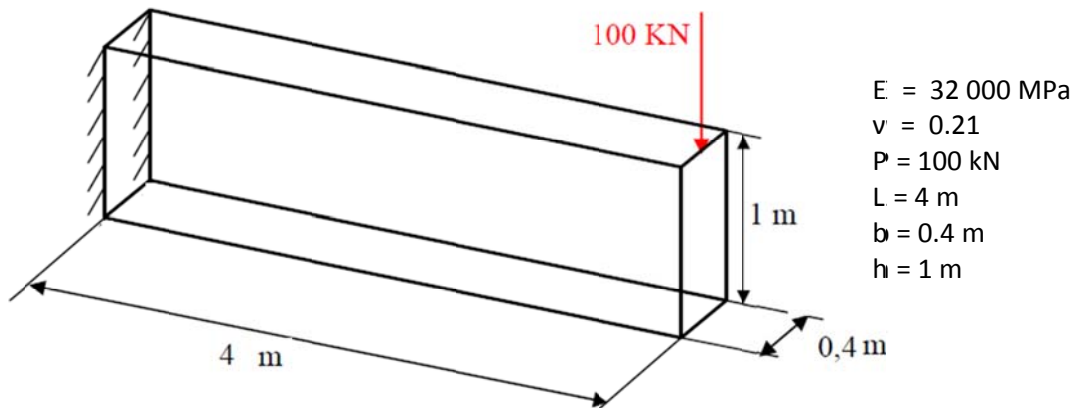


Figure 110 : exemple test 01 Poutre consol

Après maillage avec une division de la poutre en 18 unités suivant X et 5 unités suivant Y

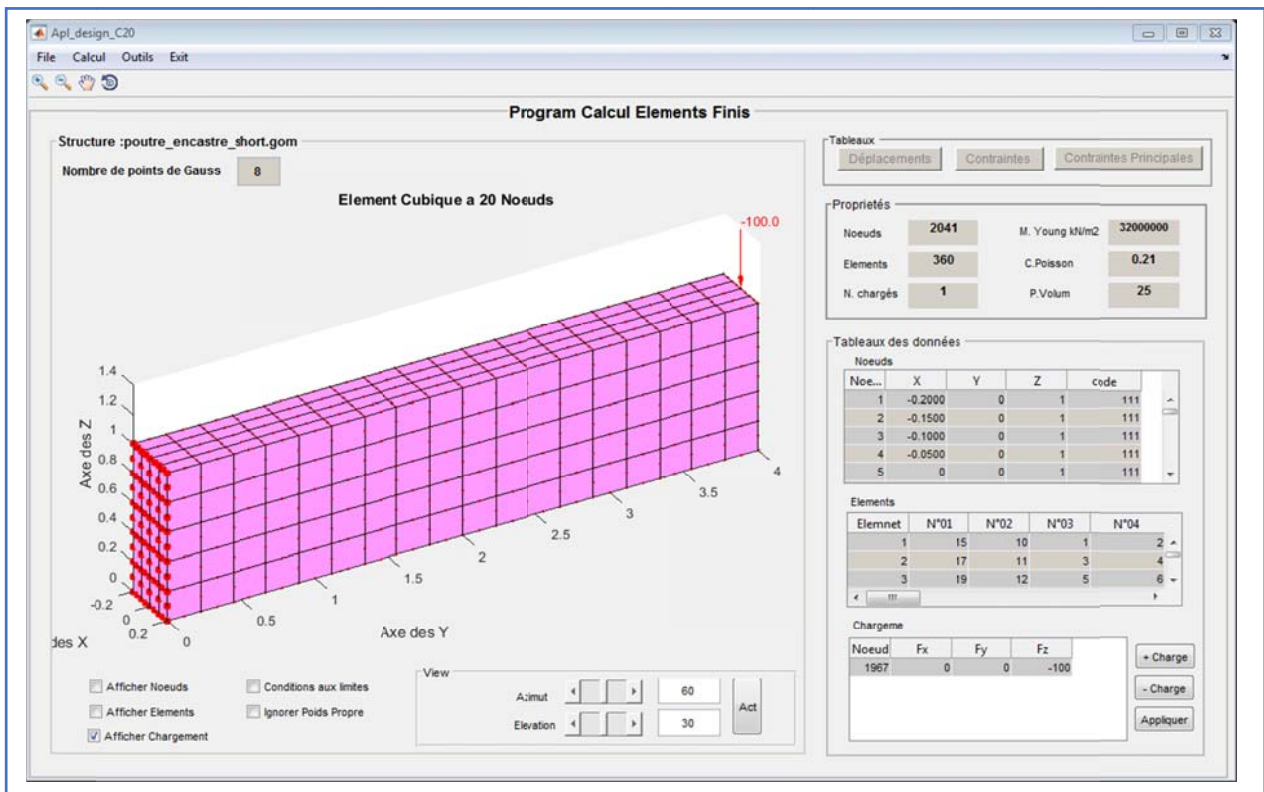


Figure 111 : maillage de la structure

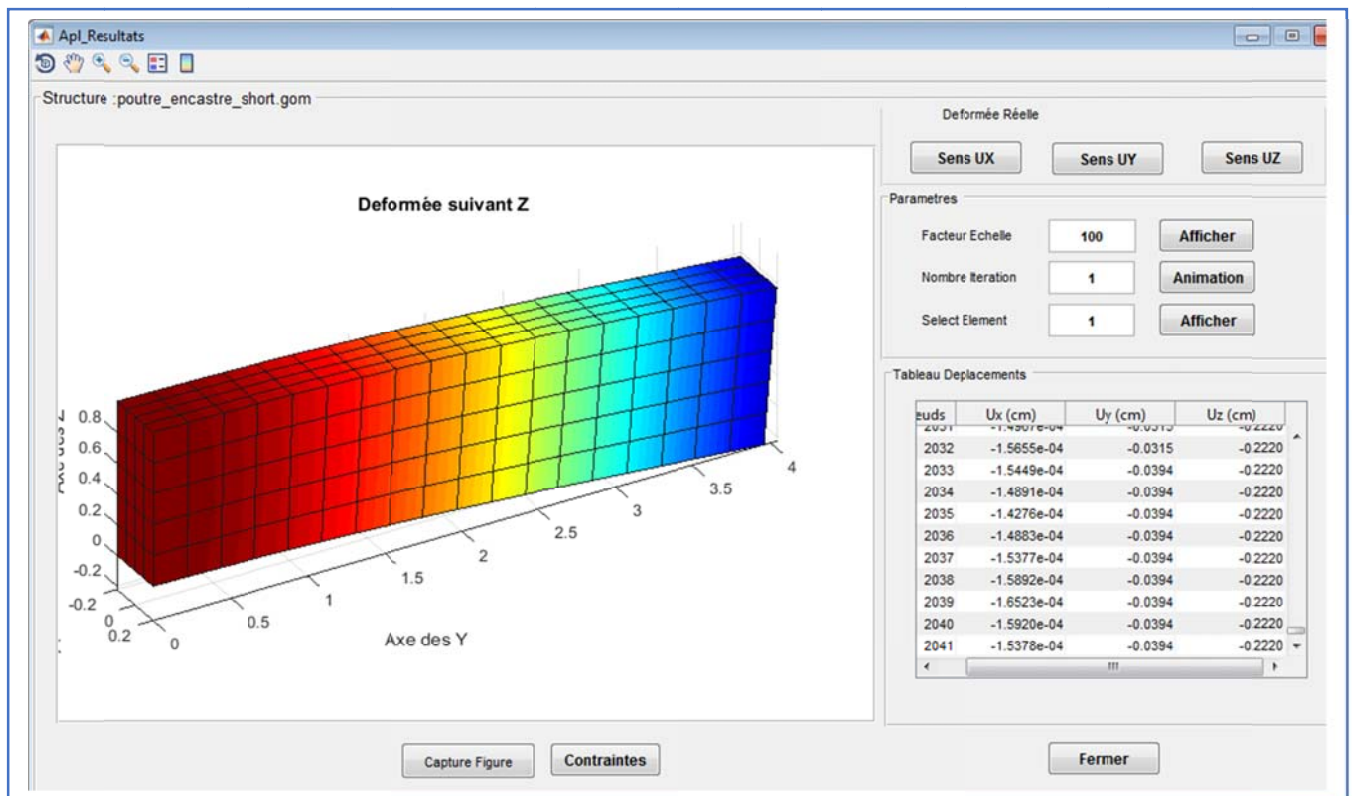


Figure 112: déformée de la structure

Les résultats obtenus sont comme suit

Déplacements dans la structure						
Déplacement_Noeuds (cm)			déplacement Max (cm)			
	coord_X	coord_Y	coord_Z	deplc_X	deplc_Y	deplc_Z
2017	0.1000	4	0.3000	-1.3320e-04	-0.0158	-0.2221
2018	0.2000	4	0.3000	-1.2318e-04	-0.0158	-0.2221
2019	-0.2000	4	0.2000	-1.7006e-04	-0.0236	-0.2220
2020	-0.1500	4	0.2000	-1.6230e-04	-0.0236	-0.2220
2021	-0.1000	4	0.2000	-1.5668e-04	-0.0236	-0.2220
2022	-0.0500	4	0.2000	-1.5236e-04	-0.0236	-0.2220
2023	0	4	0.2000	-1.5071e-04	-0.0236	-0.2220
2024	0.0500	4	0.2000	-1.4911e-04	-0.0236	-0.2220
2025	0.1000	4	0.2000	-1.4482e-04	-0.0236	-0.2220
2026	0.1500	4	0.2000	-1.3919e-04	-0.0236	-0.2220
2027	0.2000	4	0.2000	-1.3143e-04	-0.0236	-0.2220
2028	-0.2000	4	0.1000	-1.4797e-04	-0.0315	-0.2220
2029	-0.1000	4	0.1000	-1.5499e-04	-0.0315	-0.2220
2030	0	4	0.1000	-1.5238e-04	-0.0315	-0.2220
2031	0.1000	4	0.1000	-1.4967e-04	-0.0315	-0.2220
2032	0.2000	4	0.1000	-1.5655e-04	-0.0315	-0.2220
2033	-0.2000	4	0	-1.5449e-04	-0.0394	-0.2220
2034	-0.1500	4	0	-1.4891e-04	-0.0394	-0.2220
2035	-0.1000	4	0	-1.4276e-04	-0.0394	-0.2220
2036	-0.0500	4	0	-1.4883e-04	-0.0394	-0.2220
2037	0	4	0	-1.5377e-04	-0.0394	-0.2220
2038	0.0500	4	0	-1.5892e-04	-0.0394	-0.2220
2039	0.1000	4	0	-1.6523e-04	-0.0394	-0.2220
2040	0.1500	4	0	-1.5920e-04	-0.0394	-0.2220
2041	0.2000	4	0	-1.5378e-04	-0.0394	-0.2220

Figure 113 : résultats des déplacements

- Pour le point d'extrémité de coordonnées $x = 0$ m et $y=4$ $z=0$ La valeur du déplacement obtenue par **MARS** est de
 $U_x = -0.0000015377$ m
 $U_y = -0.000394$ m
 $U_z = -0.00222$ m
- Et pour le point d'extrémité de coordonnées $x = -0.2$ m $y=0$ et $z=1$, La valeur des contraintes obtenue par **MARS** est de
 $\sigma_y = -5.9645$ Mpa

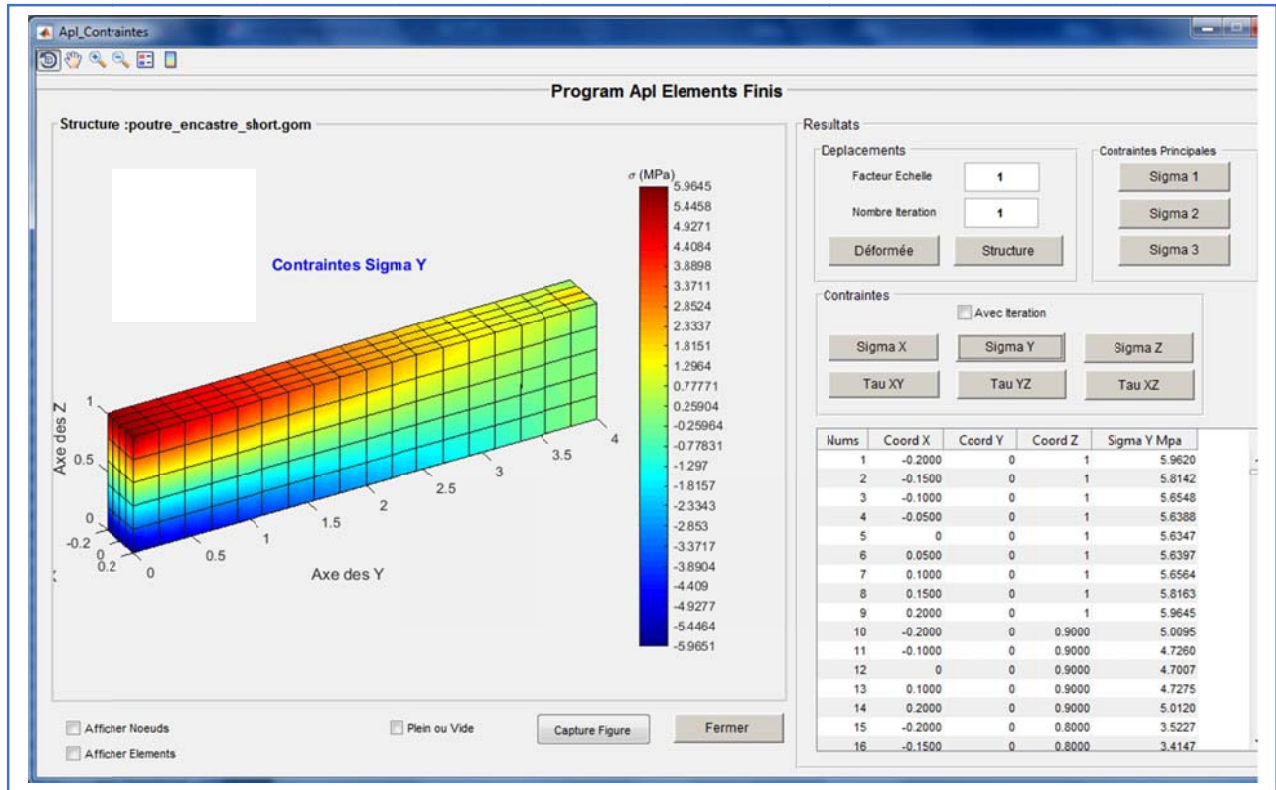


Figure 114 : distribution des contraintes σ_y

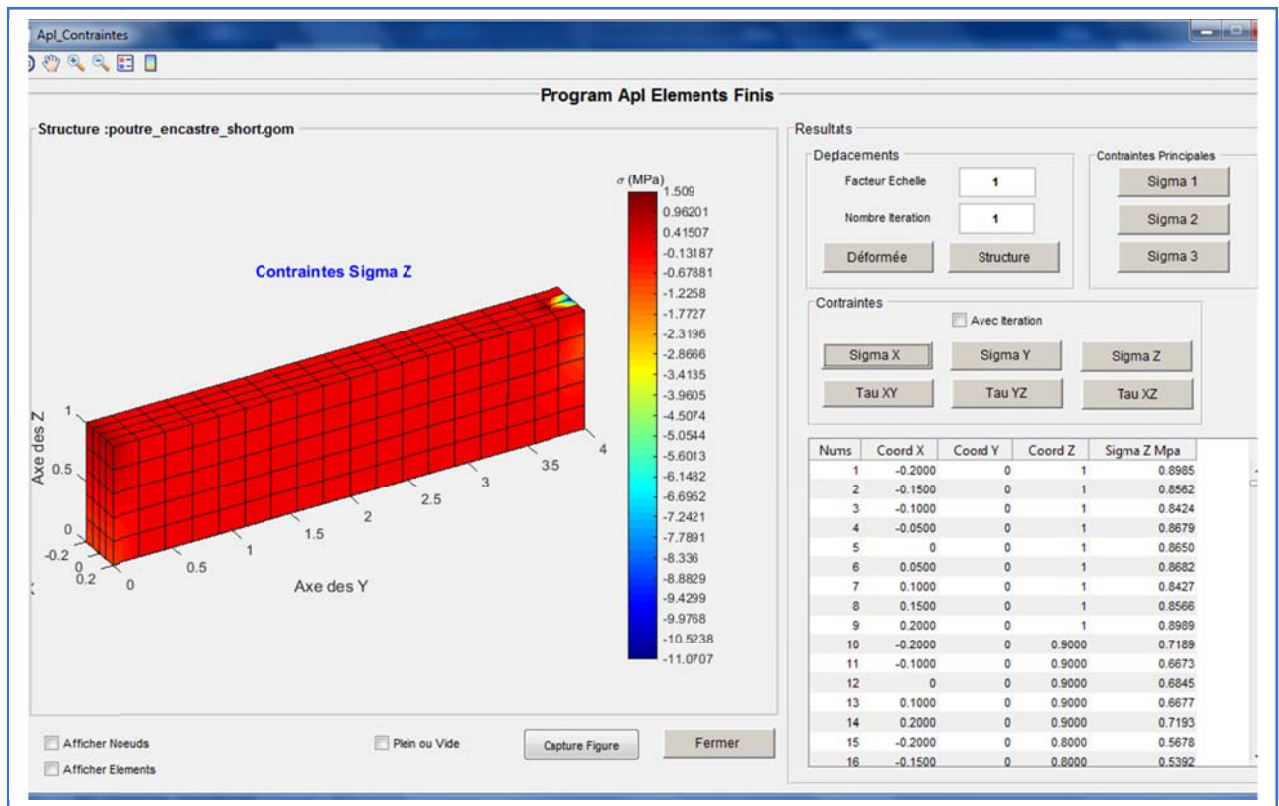


Figure 115 : distribution des contraintes σ_z

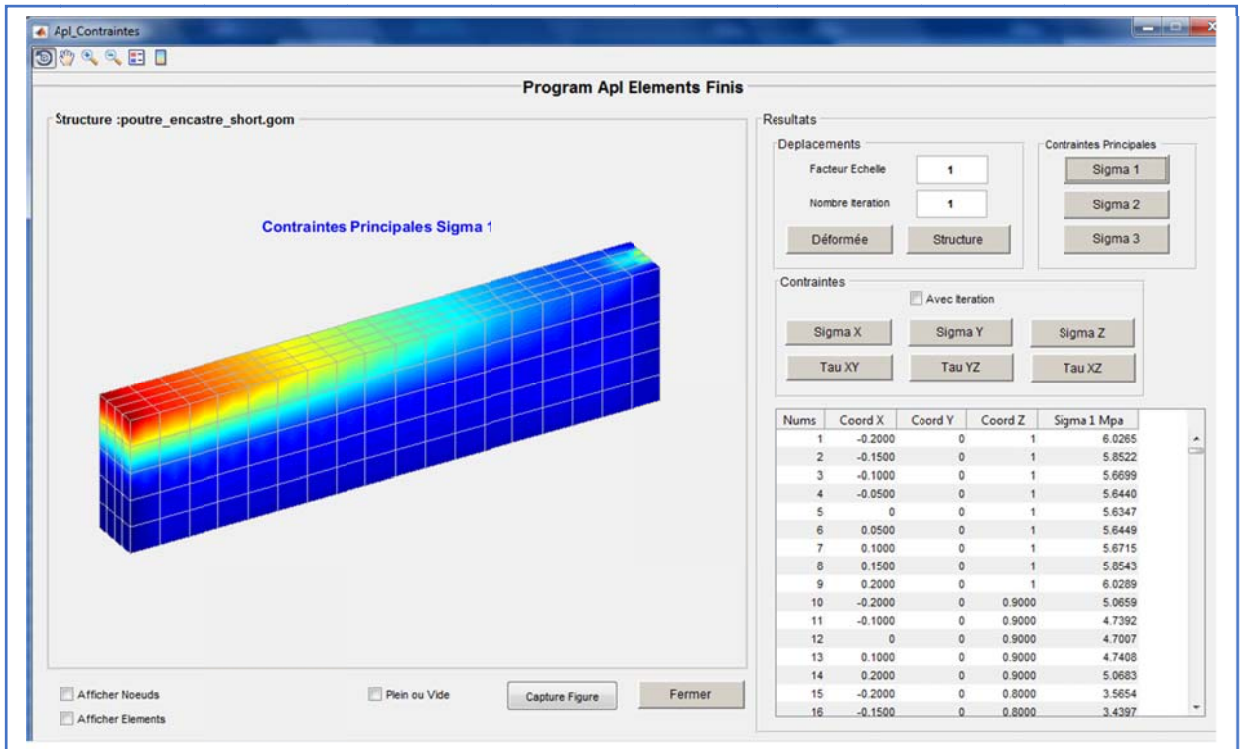


Figure 116 : distribution des contraintes σ principale 1

Contraintes dans la structure

	Contrainte_Noeuds (MPa)		Contrainte_PGauss (MPa)		Contrainte_centre_PG (MPa)	
	Sig_X	Sig_Y	Sig_Z	Tau_XY	Tau_XZ	Tau_YZ
1	7.8922e-04	0.0065	9.8934e-04	6.4099e-04	2.9252e-05	-5.9617e-04
2	7.5035e-04	0.0064	9.4321e-04	4.8446e-04	1.4470e-05	-6.3568e-04
3	8.8826e-04	0.0062	9.2845e-04	2.9522e-04	3.7319e-06	-6.9242e-04
4	0.0011	0.0062	9.5659e-04	1.6939e-04	3.3387e-06	-7.1765e-04
5	0.0011	0.0062	9.5340e-04	1.3432e-07	-5.4907e-08	-7.2570e-04
6	0.0011	0.0062	9.5686e-04	-1.6909e-04	-3.4094e-06	-7.1779e-04
7	8.884e-04	0.0062	9.2882e-04	-2.9488e-04	-3.7600e-06	-6.9270e-04
8	7.5052e-04	0.0064	9.4356e-04	-4.8422e-04	-1.4495e-05	-6.3612e-04
9	7.8946e-04	0.0065	9.8975e-04	-6.4085e-04	-2.9277e-05	-5.9670e-04
10	6.3638e-04	0.0055	7.9032e-04	5.5109e-04	3.0696e-05	-4.1709e-04
11	7.1368e-04	0.0052	7.3423e-04	2.5425e-04	6.0954e-06	-5.1184e-04
12	8.6775e-04	0.0052	7.5308e-04	8.8076e-08	-4.4840e-08	-5.4315e-04
13	7.1425e-04	0.0052	7.3459e-04	-2.5405e-04	-6.1188e-06	-5.1210e-04
14	6.3662e-04	0.0055	7.9074e-04	-5.5106e-04	-3.0717e-05	-4.1758e-04
15	4.4112e-04	0.0038	6.2243e-04	4.0203e-04	2.6368e-05	-1.8620e-04
16	4.1228e-04	0.0037	5.9116e-04	3.0292e-04	1.4604e-05	-2.2621e-04
17	4.8672e-04	0.0036	5.7716e-04	1.8639e-04	7.8259e-06	-2.8255e-04
18	5.9201e-04	0.0036	5.9247e-04	1.0825e-04	7.8070e-06	-3.0567e-04
19	5.9364e-04	0.0036	5.9045e-04	2.3657e-08	-3.4610e-08	-3.1248e-04
20	5.9217e-04	0.0036	5.9263e-04	-1.0826e-04	-7.8546e-06	-3.0578e-04
21	4.8731e-04	0.0036	5.7755e-04	-1.8639e-04	-7.8538e-06	-2.8278e-04
22	4.1247e-04	0.0037	5.9157e-04	-3.0295e-04	-1.4628e-05	-2.2656e-04
23	4.4139e-04	0.0039	6.2293e-04	-4.0214e-04	-2.6387e-05	-1.8661e-04
24	3.0533e-04	0.0025	5.1268e-04	2.6800e-04	2.1050e-05	-1.0200e-04
25	3.3404e-04	0.0023	4.8279e-04	1.2458e-04	7.5297e-06	-2.0201e-04

Figure 117 : résultats contraintes

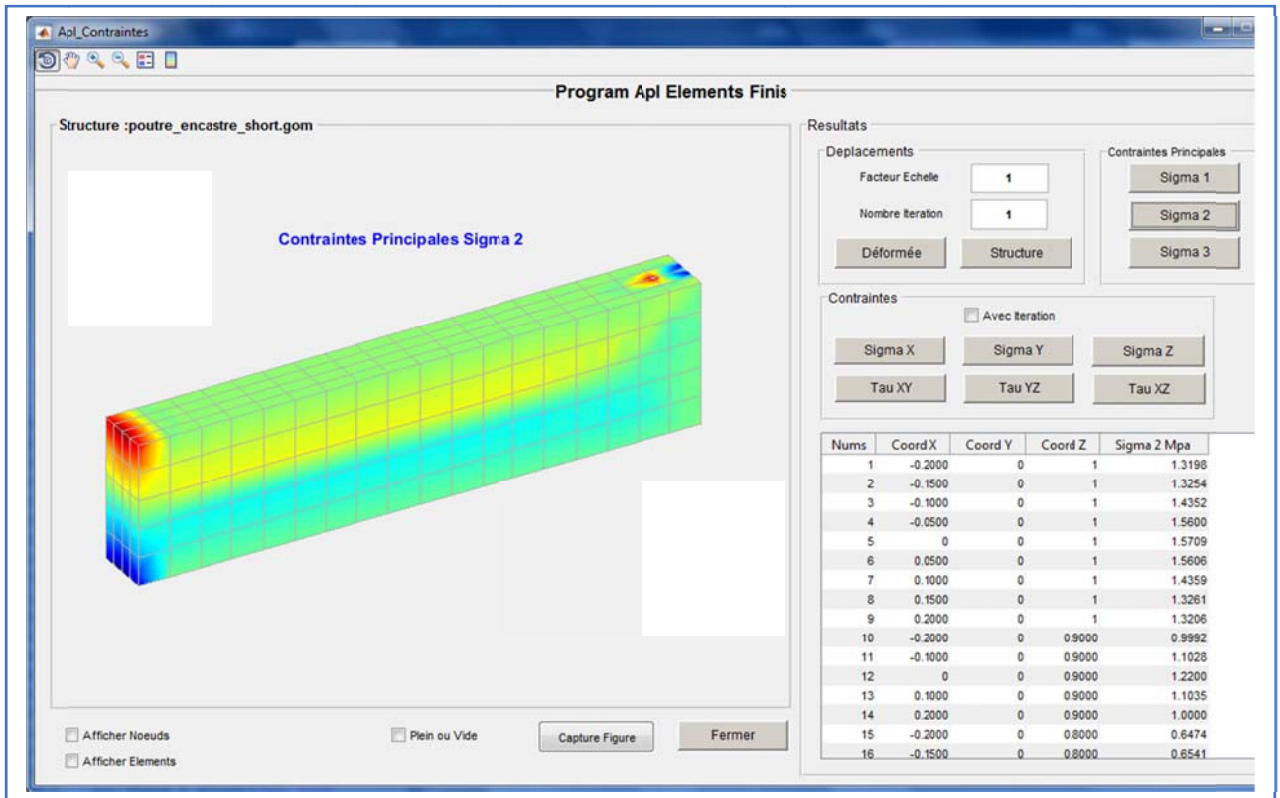


Figure 118 : distribution des contraintes σ principale 2

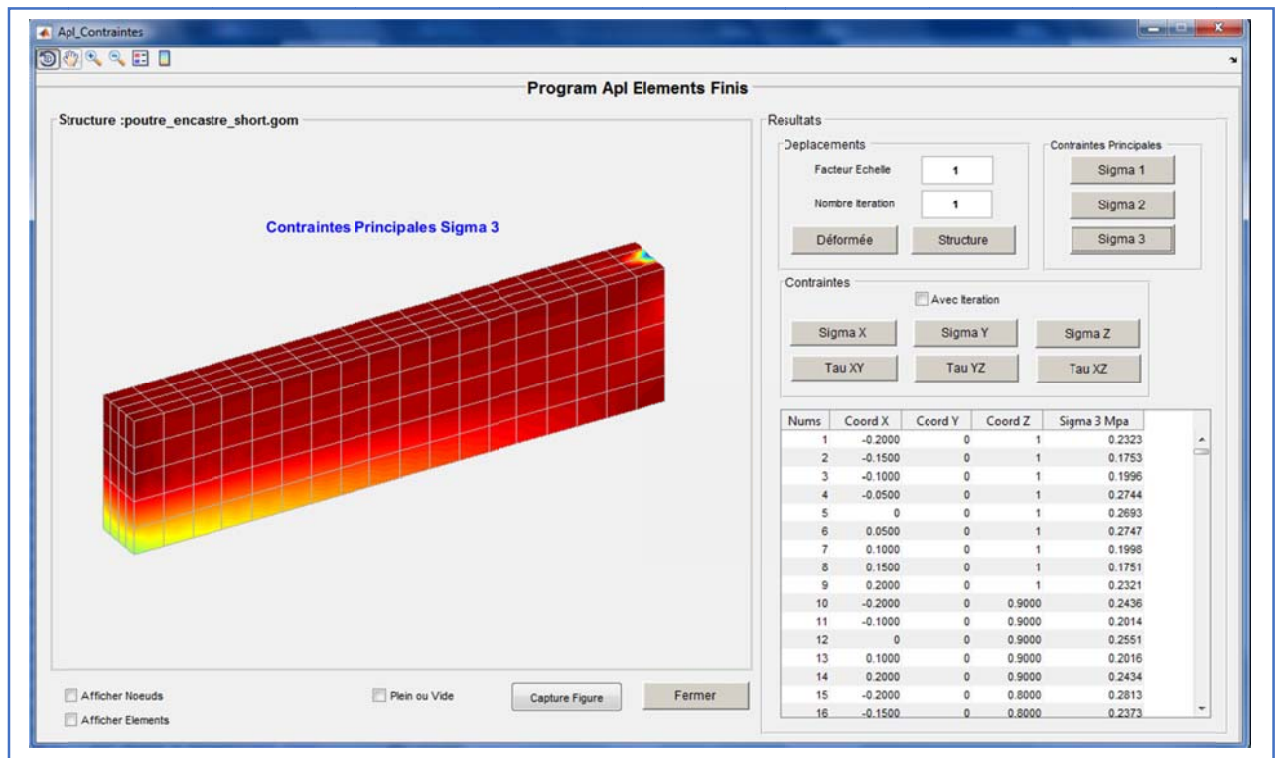


Figure 119 : distribution des contraintes σ principale 3

Test 02

Soit à étudier la structure maillée en Elément iso paramétrique Cubique à 20 nœuds données dans l'ouvrage I. M. Smith, D. V. Griffiths, L. Margetts - programming the finite element method Ref [1]

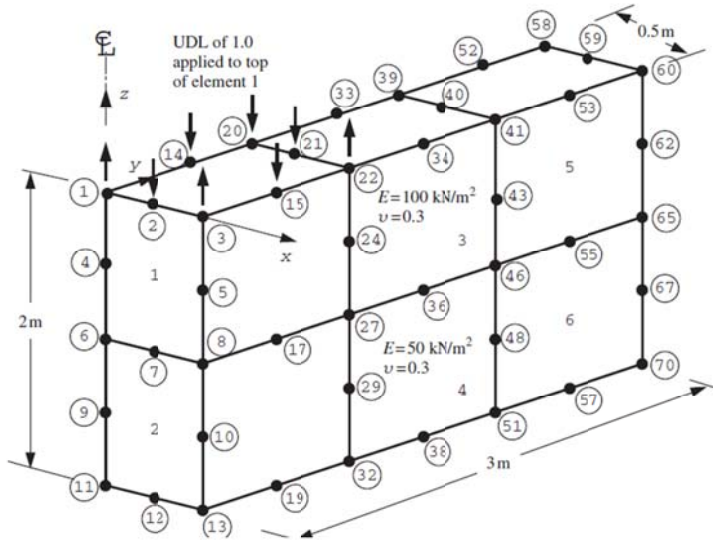


Figure 120 : exemple test 02

Structure : Mesh70inv_Brick20N.gom
 Nombre de points de Gauss : 8
 Elément Cubique a 20 Nœuds

Tableaux

Déplacements Contraintes Contraintes Principales

Propriétés

Noeuds	70	M. Young GPa	100
Elements	6	C. Poisson	0.3
N. chargés	8	P.Volum	0

Tableaux des données

Noeuds

Noe...	X	Y	Z	code
1	0	0	0	110
2	0.2500	0	0	10
3	0.5000	0	0	10
4	0	0	-0.5000	110
5	0.5000	0	-0.5000	10

Elements

Elemnet	N°01	N°02	N°03	N°04
1	6	4	1	2
2	11	9	6	7
3	25	23	20	21

Chargoeme

Noeud	Fx	Fy	Fz
1	0	0	0.0417
2	0	0	-0.1667
3	0	0	0.0417
14	0	0	-0.1667

Figure 121 : maillage de la structure

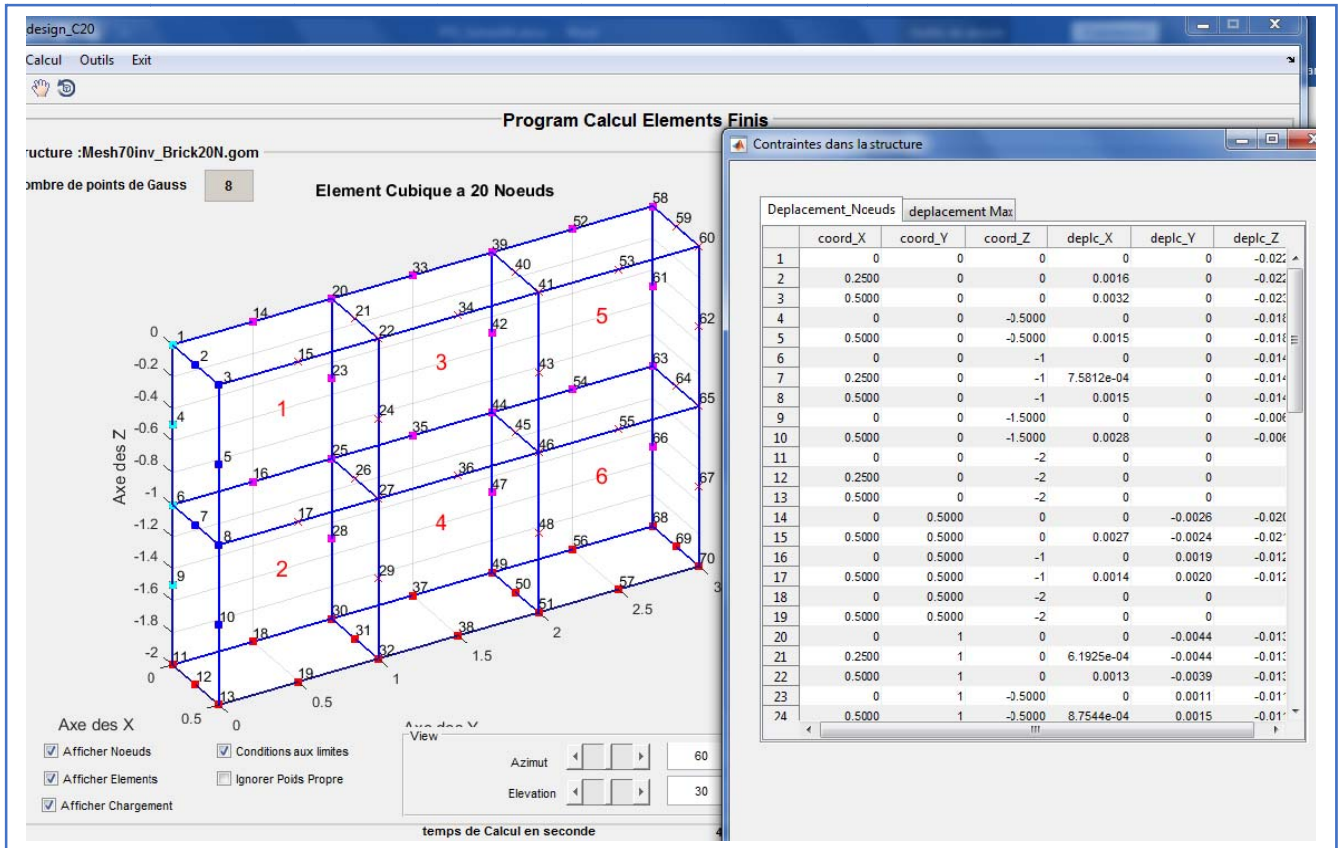


Figure 122 : maillage de structure et affichage des résultats

Node	x-disp	y-disp	z-disp
1	0.0000E+00	0.0000E+00	-0.2246E-01
2	0.1584E-02	0.0000E+00	-0.2255E-01
3	0.3220E-02	0.0000E+00	-0.2333E-01
4	0.0000E+00	0.0000E+00	-0.1849E-01
5	0.1544E-02	0.0000E+00	-0.1884E-01
6	0.0000E+00	0.0000E+00	-0.1443E-01
7	0.7581E-03	0.0000E+00	-0.1435E-01
8	0.1511E-02	0.0000E+00	-0.1411E-01
9	0.0000E+00	0.0000E+00	-0.6164E-02
10	0.2792E-02	0.0000E+00	-0.6430E-02
66	0.0000E+00	0.1572E-02	-0.1028E-03
67	-0.7448E-04	0.1716E-02	-0.5846E-04
68	0.0000E+00	0.0000E+00	0.0000E+00
69	0.0000E+00	0.0000E+00	0.0000E+00
70	0.0000E+00	0.0000E+00	0.0000E+00

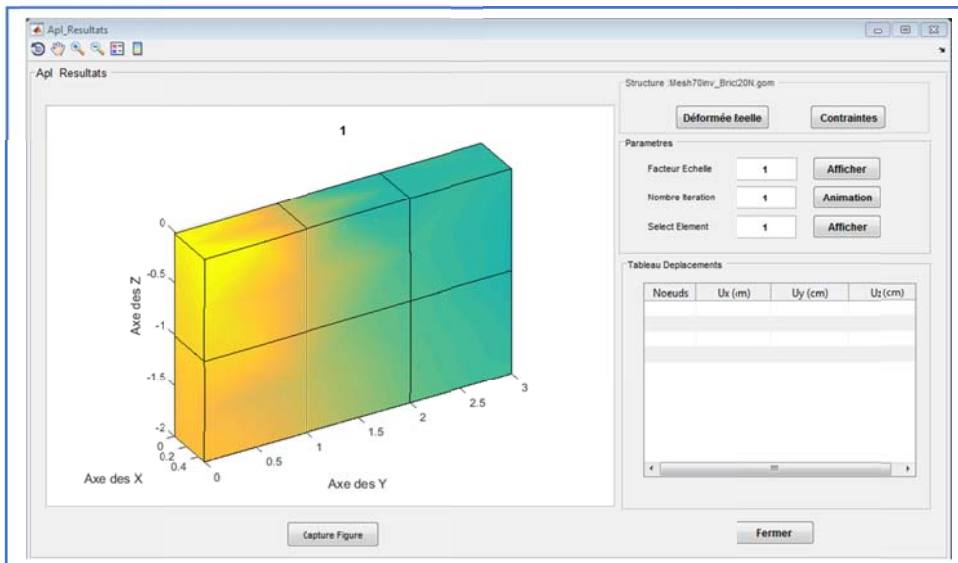


Figure 123 : déformée de la structure

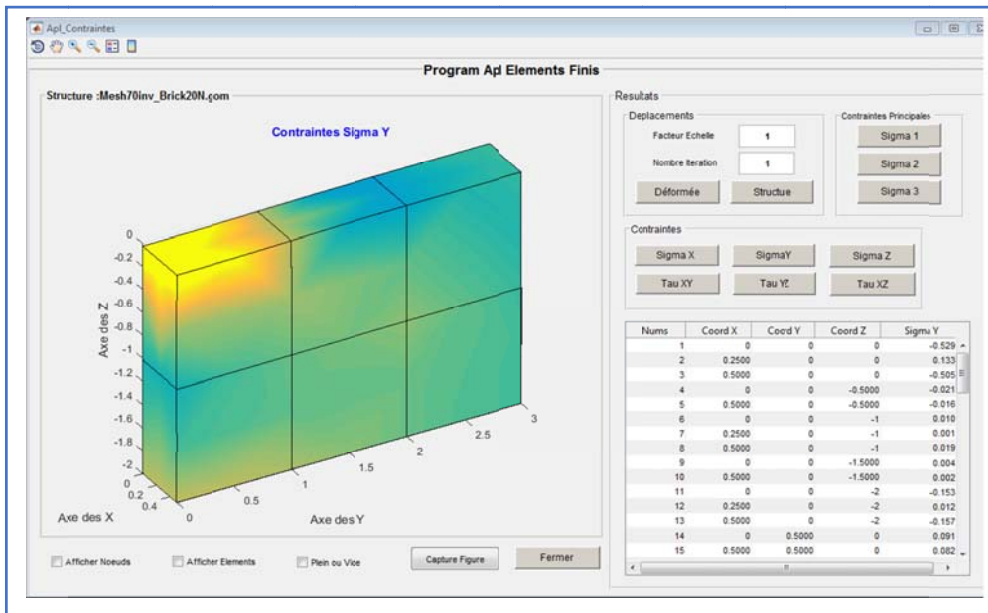


Figure 124 distribution des contraintes σ_y

Contraintes dans l'element

Contrainte_Noeuds	Contrainte_PGauss			Contrainte_centre_PG		
	Sig_X	Sig_Y	Sig_Z	Tau_XY	Tau_YZ	Tau_XZ
6	0.0121	0.0102	-0.8275	0.0012	-0.0098	0.0528
4	-2.9694e-04	-0.0210	0.4284	6.0322e-07	-3.1750e-06	-9.1896e-04
1	-0.0152	-0.5294	-1.0128	4.0545e-04	0.0153	-0.0253
2	2.1895e-05	0.1337	0.5348	5.9916e-07	1.0188e-04	2.2703e-04
3	-0.0029	-0.5052	-1.0561	0.0030	0.0133	-0.0179
5	-1.8972e-05	-0.0168	0.4917	1.3647e-05	3.5658e-06	-6.8695e-04
8	0.0114	0.0195	-0.8436	0.0022	-0.0197	0.0646
7	9.3003e-05	0.0017	0.3499	7.5576e-06	1.9054e-04	0.0021
16	2.1497e-04	0.0012	0.2686	-7.4404e-08	7.4684e-05	0.0056
14	2.1708e-04	0.0914	0.3816	-1.8379e-06	3.7692e-05	-0.0019
15	1.9271e-05	0.0824	0.4095	-1.4673e-05	6.9437e-05	-0.0015
17	4.4859e-05	2.2310e-04	0.2797	1.7716e-05	3.1460e-04	0.0060
25	0.0051	-0.0804	-0.5071	1.4897e-04	-0.0069	0.1698
23	-9.8020e-05	0.0014	0.1444	1.3250e-06	-1.0609e-06	0.0159
20	-0.0084	-0.1962	-0.4904	-0.0090	0.0054	0.1470
21	1.2381e-04	0.0291	0.1601	5.1829e-05	1.5394e-05	0.0120
22	-0.0019	-0.2017	-0.5115	-0.0115	0.0061	0.1537
24	-7.3373e-06	0.0037	0.1634	-3.5534e-05	1.2905e-05	0.0190
27	0.0074	-0.0778	-0.5092	0.0011	-0.0134	0.1762
26	-1.4627e-05	0.0044	0.1406	-1.0322e-06	9.3604e-05	0.0163

Figure 125 : Résultats des déplacements

Figure 126 : résultats de comparaison

The integration point (nip= 1) stresses are:

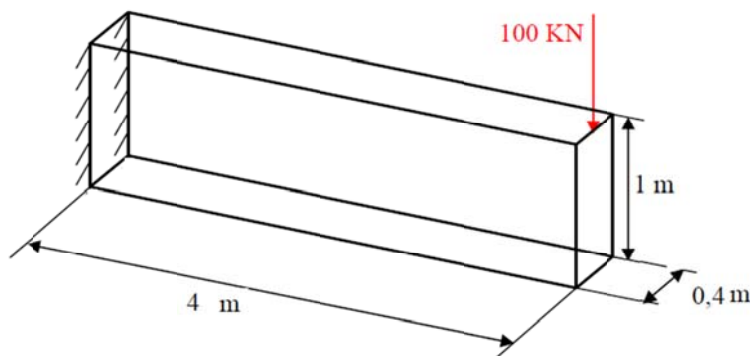
Element	x-coord	y-coord	z-coord	tau_yz	tau_zx
1	0.2500E+00	0.5000E+00	-0.5000E+00	0.9597E-01	0.4352E-02
2	0.2500E+00	0.5000E+00	-0.1500E+01	0.7614E-01	0.4169E-02
3	0.2500E+00	0.1500E+01	-0.5000E+00	0.9399E-01	-0.2814E-02
4	0.2500E+00	0.1500E+01	-0.1500E+01	0.1214E+00	0.2939E-02
5	0.2500E+00	0.2500E+01	-0.5000E+00	-0.8733E-02	0.8652E-03
6	0.2500E+00	0.2500E+01	-0.1500E+01	0.6083E-01	0.5078E-04

Ref [1]

4.8 Apl design Cub8

Test 01

On se propose d'étudier la flexion d'une poutre encastree a l'une de ses extremités et libre de l'autre, soumise a une charge concentree $P=100$ KN, de longueur $L = 4.00$ m, de section rectangulaire de largeur 0.4 m, avec une hauteur de 1 m, et sans tenir compte du poids propre de la structure



$E = 32\ 000$ MPa
 $\nu = 0.21$
 $P = 100$ kN
 $L = 4$ m
 $b = 0.4$ m
 $h = 1$ m

Figure 127 : exemple test 01 Poutre consol

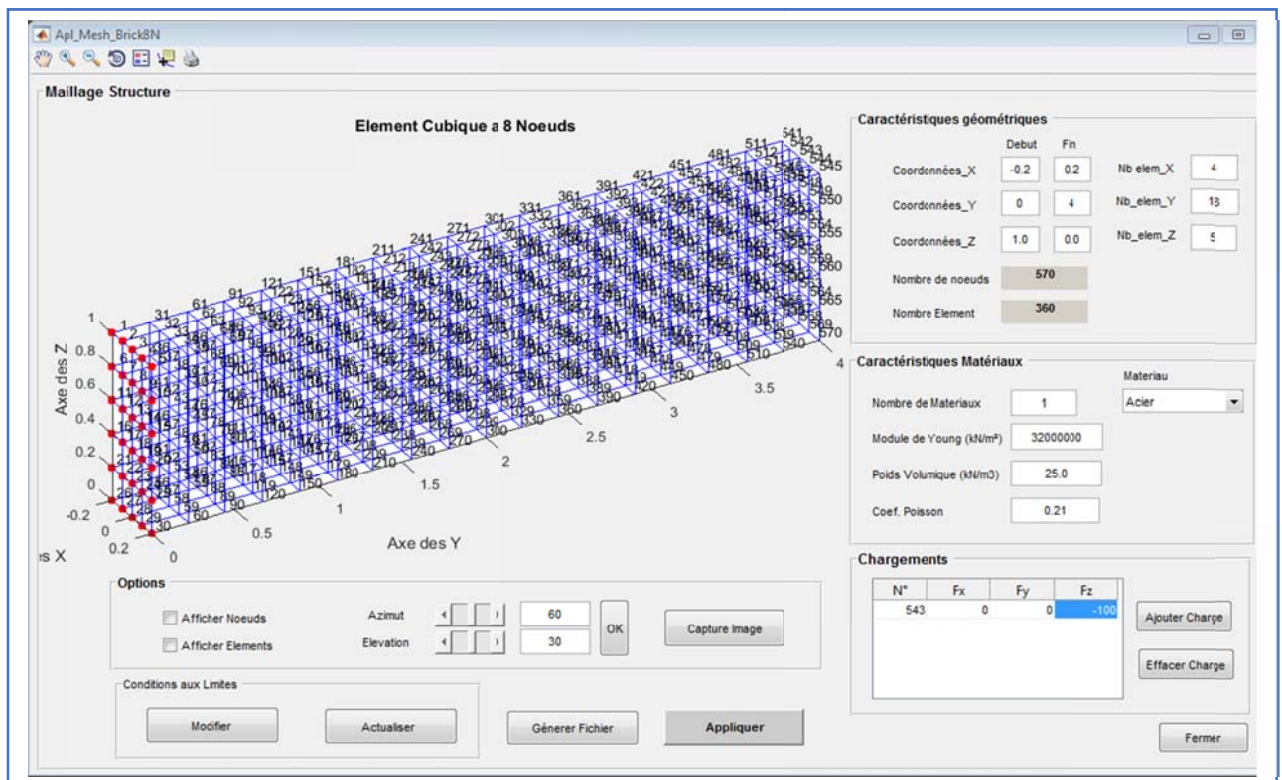


Figure 128 : maillage de la structure avec Apl_mesh_tetredre

après maillage et ajout des condition aux limites

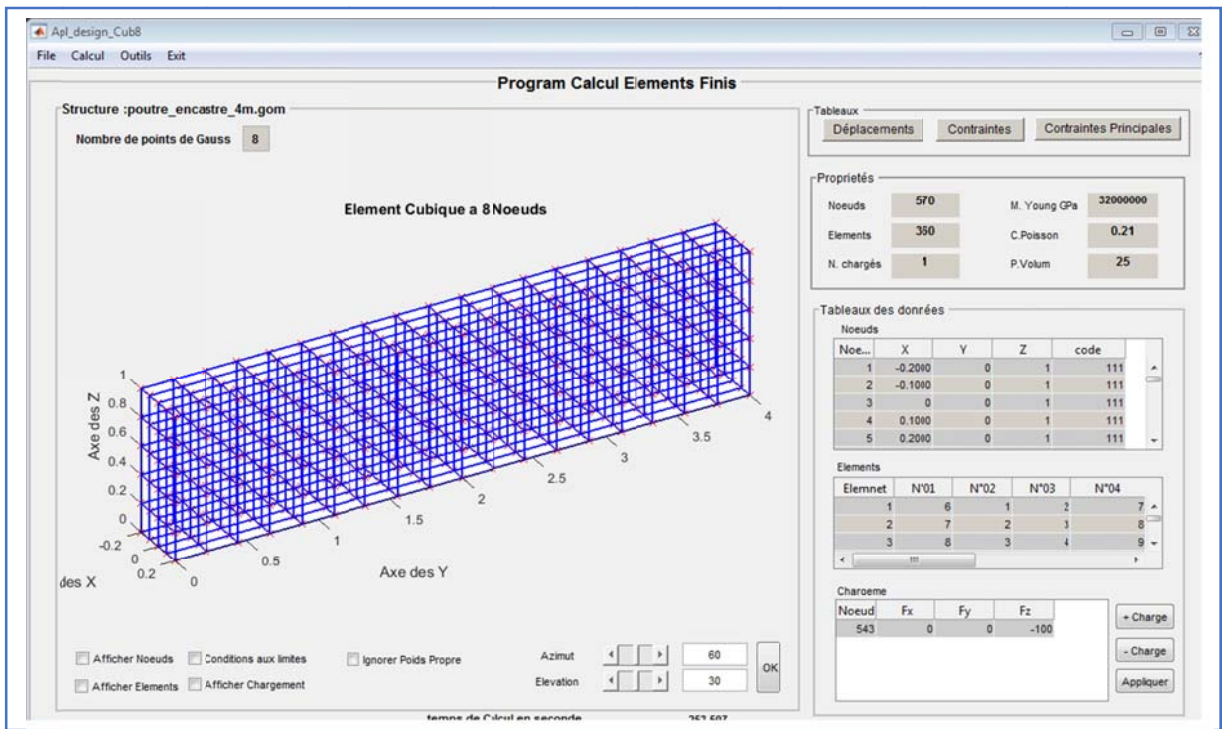


Figure 129 : Affichage de la structure

Déplacement

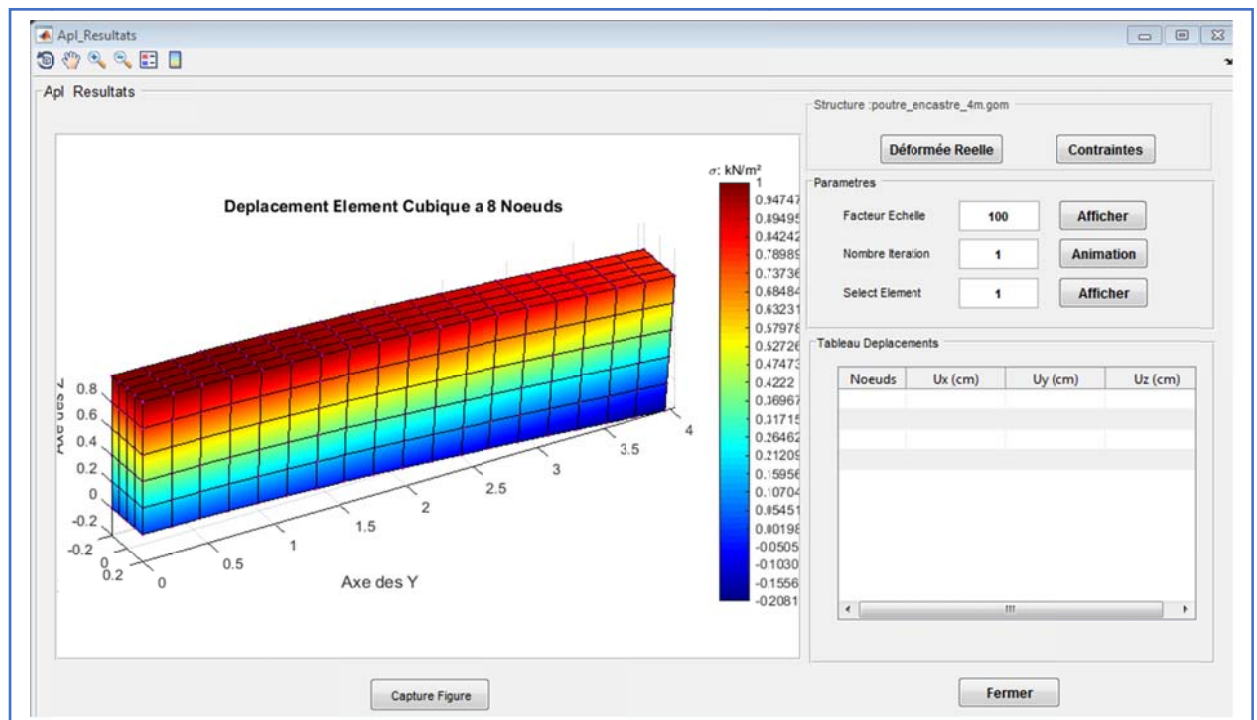


Figure 130 déformée de la structure

Le point d'extrémité pour la comparaison est le point situé aux coordonnées $x=0$; $y=4$; $z=0$ (remarque notre poutre est située dans le plan yz)

561	-0.20	4.00	0.20	-2.18917307...	-2.229907332000000e-04	-0.002081454482000
562	-0.10	4.00	0.20	0	-2.230354288000000e-04	-0.002081367706000
563	0.00	4.00	0.20	0	-2.230441435000000e-04	-0.002081334048000
564	0.10	4.00	0.20	0	-2.230354288000000e-04	-0.002081367706000
565	0.20	4.00	0.20	2.189173035...	-2.229907332000000e-04	-0.002081454482000
566	-0.20	4.00	0.00	0	-3.719219609000000e-04	-0.002081257653000
567	-0.10	4.00	0.00	0	-3.719806853000000e-04	-0.002081180448000
568	0.00	4.00	0.00	0	-3.720032990000000e-04	-0.002081165357000
569	0.10	4.00	0.00	0	-3.719806853000000e-04	-0.002081180448000
570	0.20	4.00	0.00	0	-3.719219609000000e-04	-0.002081257653000

Figure 131 Résultats des déplacements

Contraintes Tau XZ

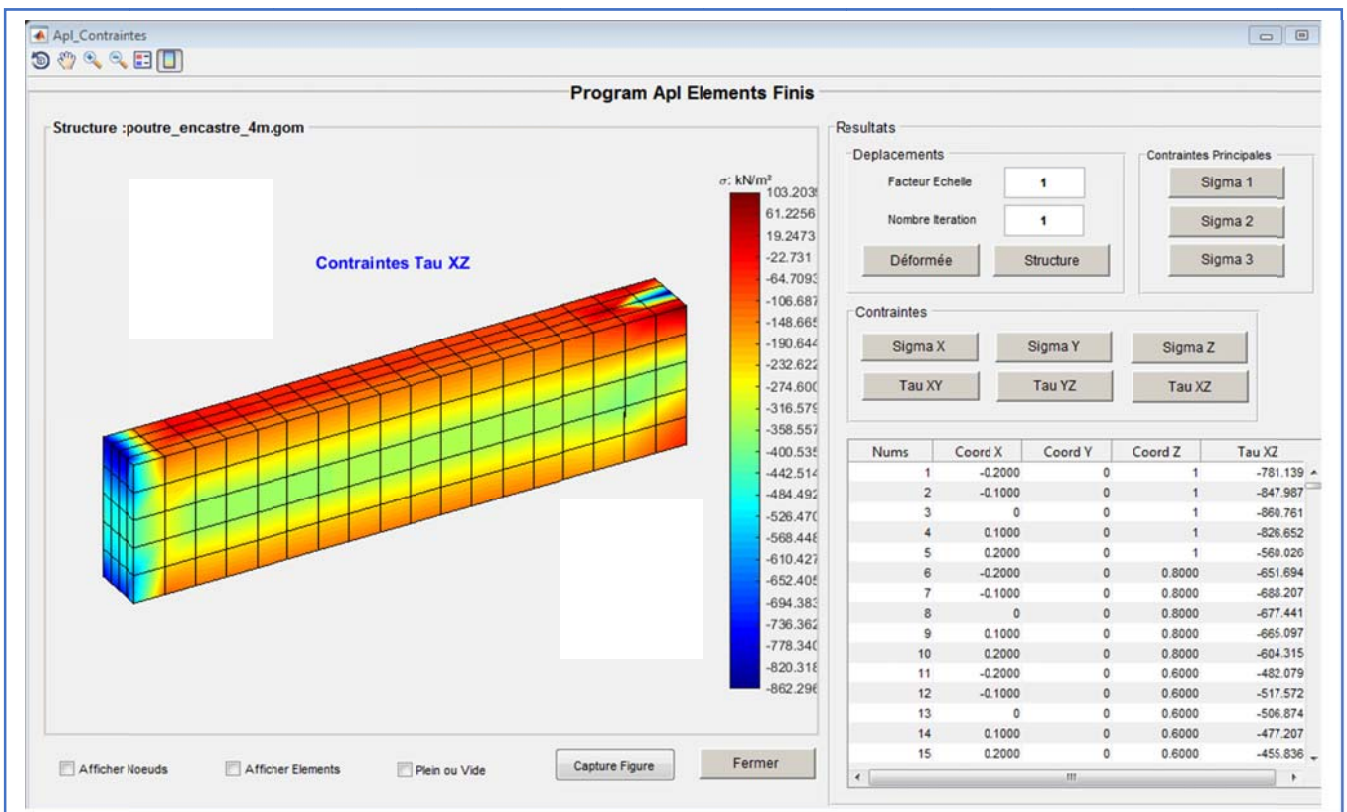


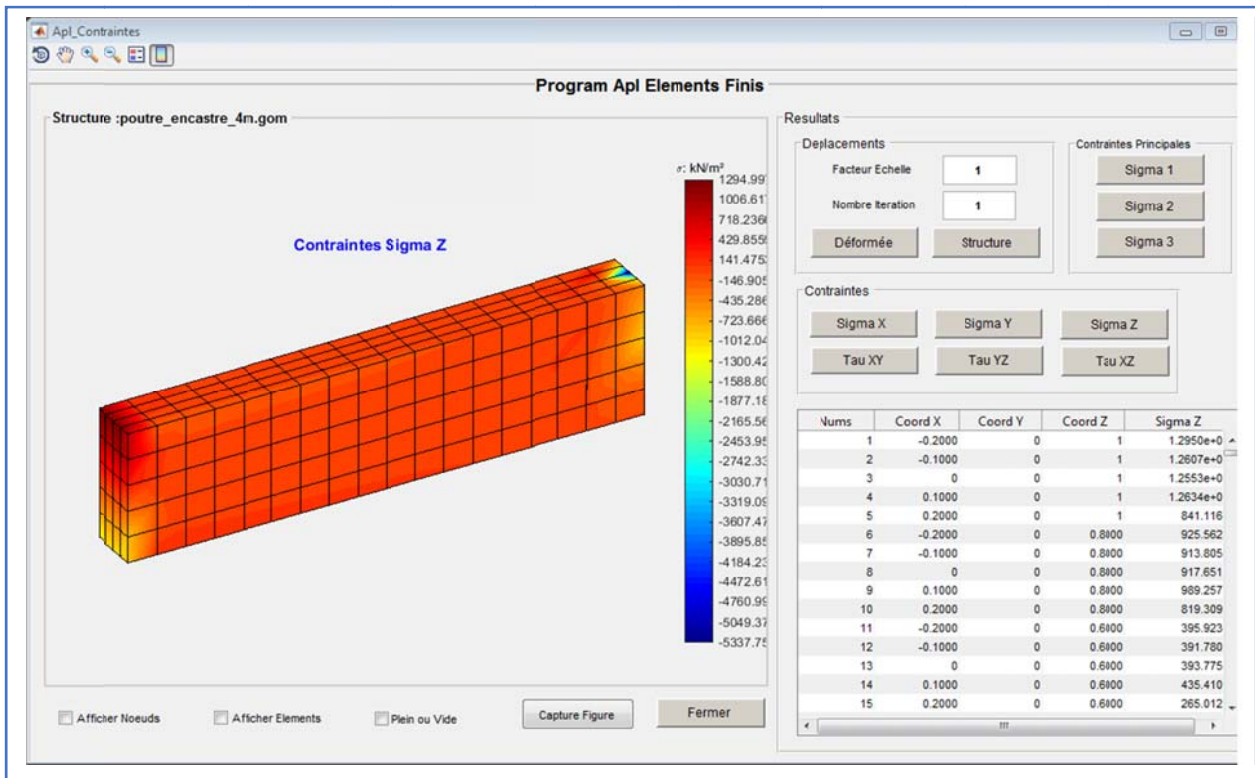
Figure 132 : distribution des contraintes

Contraintes dans la structure

Contrainte_Noeuds	Contrainte_PGauss			Contrainte_centre_PG		
	Sig_X	Sig_Y	Sig_Z	Tau_XY	Tau_YZ	Tau_XZ
1	1.2432e+03	5.9597e+03	1.2950e+03	376.7668	9.4737	-781.1394
2	1.2420e+03	5.8155e+03	1.2607e+03	165.0354	7.5294	-847.9875
3	1.2377e+03	5.8006e+03	1.2553e+03	-41.0526	5.6228	-860.7613
4	1.2136e+03	5.8502e+03	1.2634e+03	-234.3763	6.0504	-826.6529
5	851.4262	4.1902e+03	841.1181	-274.7745	-11.4218	-560.0261
6	899.1849	4.3304e+03	925.5627	186.5246	3.8024	-651.6942
7	903.6630	4.2775e+03	913.8055	33.9592	2.3234	-688.2075
8	899.5859	4.2935e+03	917.6518	-118.9631	0.2984	-677.4415
9	931.9923	4.4804e+03	989.2573	-251.7435	-2.1847	-665.0972
10	789.6552	3.8124e+03	819.3094	-245.5466	-9.3331	-604.3151
11	365.9356	1.7645e+03	395.9239	74.5666	4.5623	-482.0790
12	368.1269	1.7432e+03	391.7809	9.9363	2.8056	-517.5722
13	367.0559	1.7531e+03	393.7754	-55.2057	0.2531	-506.8740
14	389.2683	1.8665e+03	435.4100	-116.7486	-2.4236	-477.2076
15	248.0174	1.1964e+03	265.0126	-78.7468	-9.7103	-455.8365
16	-113.2288	-553.3521	-114.4440	-35.2429	4.5760	-448.5087
17	-111.2718	-538.2282	-110.6299	-13.3986	2.7233	-483.9575
18	-109.4722	-537.6355	-110.0990	6.8442	-0.0946	-473.5347
19	-91.4174	-475.0280	-80.4086	11.9099	-2.9550	-438.5324
20	-248.0174	-1.1964e+03	-265.0125	78.7468	-9.7103	-455.8365
21	-603.4705	-2.9314e+03	-618.3300	-145.1599	4.9366	-533.3781
22	-602.0756	-2.8807e+03	-606.0258	-37.1268	2.9097	-568.9520
23	-598.3137	-2.8904e+03	-606.9296	68.6133	0.4761	-558.1266
24	-581.6927	-2.8832e+03	-579.3247	142.4411	-2.5451	-515.3177
25	-789.6552	-3.8124e+03	-819.3094	245.5466	-9.3331	-604.3151

Figure 133 : résultats des contraintes

Contraintes Sigma Z



gure 134 : distribution des contraintes σ_z

Contraintes Sigma X

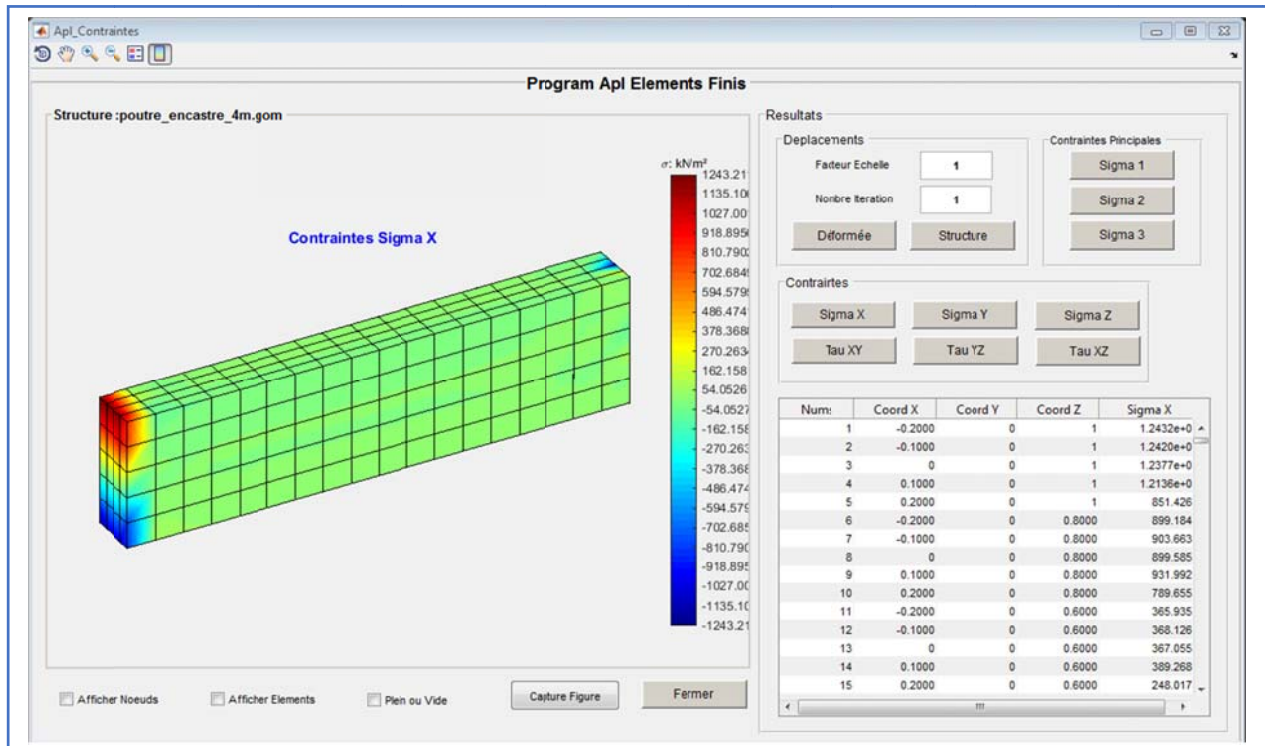
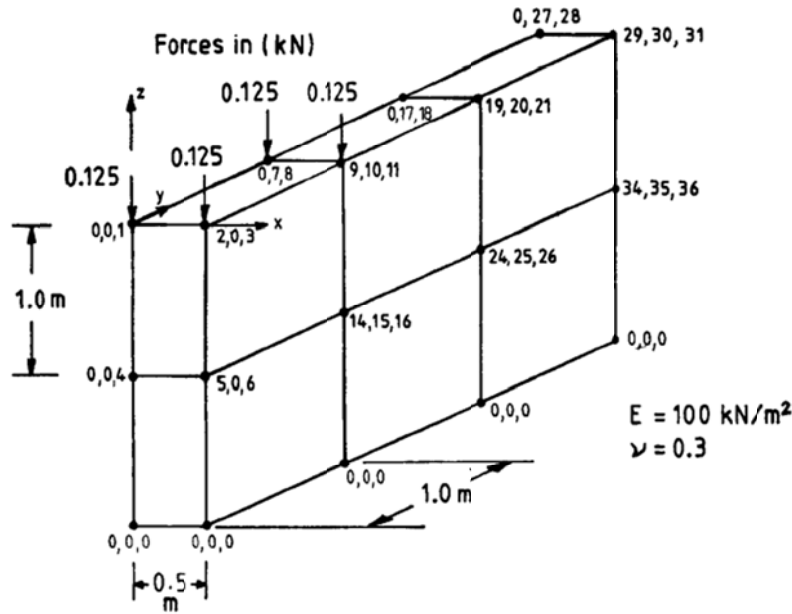


Figure 135 : distribution des contraintes σ_x

Test 02

Soit a reprendre l'exemple données par I. M. Smith, D. V. Griffiths, L. Margetts - programming the finite element method



I. M. Smith, D. V. Griffiths, L. Margetts - programming the finite element method

Figure 136 exemple test2

Structure : cube_01_Griffs.gom

Nombre de points de Gauss : 8

Element Cubique a 8 Noeuds

Propriétés

Noeuds	24	M. Young GPa	100
Elements	6	C.Poisson	0.3
N. chargés	4	P.Volum	0

Tableau des données

Noeuds

Noe...	X	Y	Z	code
1	0	0	2	110
2	0.5000	0	2	10
3	0	0	1	110
4	0.5000	0	1	10
5	0	0	0	111

Elements

Elemnet	N°01	N°02	N°03	N°04
1	3	1	2	4
2	5	3	4	6
3	9	7	8	10

Charge

Noeud	Fx	Fy	Fz
1	0	0	-0.1250
2	0	0	-0.1250
7	0	0	-0.1250
8	0	0	-0.1250

temps de Calculen seconde : 11.5204

Figure 137 maillage et affichage de la structure

Contraintes dans la structure

Deplacement_Aux_Noeuds				Deplacement Max		
	coord_X	coord_Y	coord_Z	deplc_X	deplc_Y	deplc_Z
1	0.00	0.00	2.00	0.00	0	-0.01595118...
2	0.50	0.00	2.00	0.00	0	-0.01611717...
3	0.00	0.00	1.00	0.00	0	-0.00653356...
4	0.50	0.00	1.00	0.00	0	-0.00684213...
5	0.00	0.00	0.00	0.00	0	0
6	0.50	0.00	0.00	0.00	0	0
7	0.00	1.00	2.00	0.00	-0.00153012...	-0.00903388...
8	0.50	1.00	2.00	0.00	-0.00119726...	-0.00901130...
9	0.00	1.00	1.00	0.00	0.001631524...	-0.00427170...
10	0.50	1.00	1.00	0.00	0.001819641...	-0.00443683...
11	0.00	1.00	0.00	0.00	0	0
12	0.50	1.00	0.00	0.00	0	0
13	0.00	2.00	2.00	0.00	-0.00119347...	-0.00110714...
14	0.50	2.00	2.00	-0.00	-0.00105005...	-9.66461035...
15	0.00	2.00	1.00	0.00	0.001431240...	-9.64120431...
16	0.50	2.00	1.00	0.00	0.001557348...	-9.72204135...
17	0.00	2.00	0.00	0.00	0	0
18	0.50	2.00	0.00	0.00	0	0
19	0.00	3.00	2.00	0.00	-6.39003055...	5.760290735...
20	0.50	3.00	2.00	-0.00	-6.62867703...	6.551551882...
21	0.00	3.00	1.00	0.00	8.487248827...	2.654314014...
22	0.50	3.00	1.00	0.00	8.923047701...	2.818374727...

Les résultats sont donnés comme suit a comparais avec ceux donnés par . Griffiths Ref [1]

Figure 138 résultats des déplacements

Contraintes dans la structure

Contrainte_Noeuds				Contrainte_PGauss			Contrainte_centre_PG		
	coord_X	coord_Y	coord_Z	Sig_Xgp	Sig_Ygp	Sig_Zgp	Tau_XYgp	Tau_YZgp	Tau_XZgp
1	0.2500	0.5000	1.5000	0.0056	-0.2087	-0.7617	-0.0069	-0.0044	0.1203
2	0.2500	0.5000	0.5000	-0.0951	-0.1279	-0.6190	-0.0014	0.0157	0.0781
3	0.2500	1.5000	1.5000	0.0157	-0.0708	-0.2534	-0.0032	-0.0042	0.1090
4	0.2500	1.5000	0.5000	-0.0700	-0.1305	-0.3263	-8.7843e-04	0.0096	0.1270
5	0.2500	2.5000	1.5000	0.0060	-0.0013	0.0150	0.0026	-0.0019	-0.0239
5	0.2500	2.5000	0.5000	-0.0147	-0.0520	-0.0547	6.3972e-04	0.0035	0.0694

Figure 139 : résultats des contraintes

-.1595E-01	.2297E-02	-.1612E-01	-.6534E-02	.1550E-02	-.6842E-02	} Displacements
-.1530E-02	-.9034E-02	.1056E-02	-.1197E-02	-.9011E-02	.1632E-02	
-.4272E-02	.1030E-02	.1820E-02	-.4437E-02	-.1193E-02	-.1107E-02	
-.1366E-03	-.1050E-02	-.9665E-03	.1431E-02	-.9641E-03	.3100E-03	
.1557E-02	-.9722E-03	-.6390E-03	.5760E-03	-.1735E-03	-.6629E-03	
.6552E-03	.8487E-03	.2654E-03	.3694E-04	.8924E-03	.2818E-03	
.5554E-02	-.2087E+00	-.7617E+00	-.6923E-02	.1203E+00	-.4432E-02	} Centroid stresses
-.9508E-01	-.1279E+00	-.6190E+00	-.1389E-02	.7806E-01	.1570E-01	
.1573E-01	-.7076E-01	-.2534E+00	-.3187E-02	.1090E+00	-.4232E-02	
-.7003E-01	-.1305E+00	-.3263E+00	-.8784E-03	.1270E+00	.9555E-02	
.5950E-02	-.1346E-02	.1505E-01	.2583E-02	-.2391E-01	-.1930E-02	
-.1468E-01	-.5201E-01	-.5473E-01	.5397E-03	.6936E-01	.3496E-02	
						} $\sigma_x \sigma_y \sigma_z$
						} $\tau_{xy} \tau_{yz} \tau_{zx}$

Figure 5.35 Results from Program 5.9

I. M. Smith, D. V. Griffiths, L. Margetts - programming the finite element method

Figure 140 : résultats de comparaison

4.9 Conclusion

Afin de montrer l'influence du choix du type d'éléments finis utilisé pour la discrétisation des structures étudiées sur la convergence vers la solution exacte ; six (06) études comparatives on était menées.

Etude 01 :

dans le paragraphe précédent ,cinq (05) éléments finis (Q8,T6,C4,C20 et tétraèdre T4) ont été utilisés pour discrétiser la poutre de 4 m de long en gardant au maximum le même nombre d'élément finis utilisés soit 4 dans le sens X 5 divisions dans le sens Z et 18 division dans le sens Y

Le tableau suivant résume la comparaison des résultats RDM, avec les valeurs obtenues par notre programme

Résultats	Théoriques	Résultats Apl_Design_MARS				
		Q8	T6	T4	C8	C20
sigmaX (Mpa)	6.000	5.7395	5.55317	3.9285	5.9597	5.962
W (y,z) (m)	-0,002000	0.002084	0.0020806	0.001392	0.0020811	0.00222

Tableau 13 valeurs comparatives des résultats

Dans ce contexte nous avons dressé le graphe *figure 139* qui représente l'impact du choix du types d'élément sur les résultats. Ainsi l'allure du déplacement vertical w en fonction de son abscisse tout le long de la poutre se présente comme suit

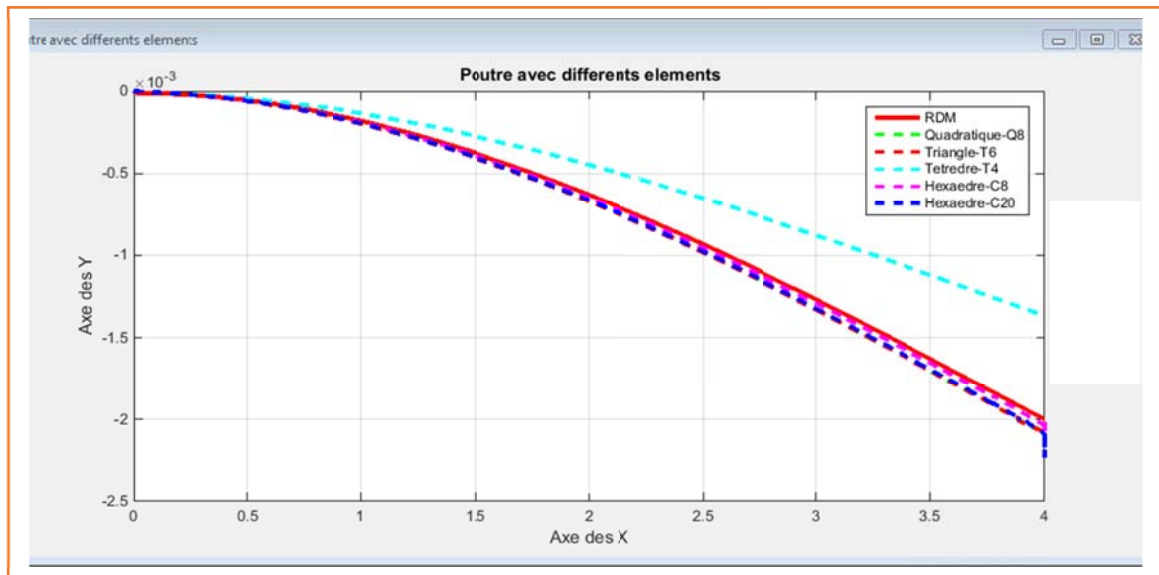


Figure 141 : l'évolution du déplacement vertical w le long de la poutre

On constate ainsi que ;

Les valeurs des déplacements et des contraintes convergent nettement vers celles obtenue par les formules RDM et cela pour tous les éléments quadratiques.

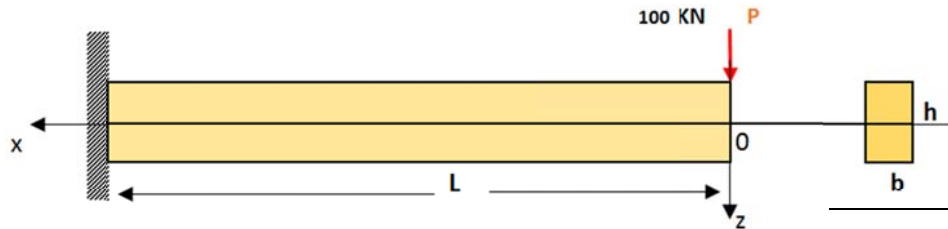
La solution obtenue par les éléments tétraèdre T4 et hexaèdre C8 linéaires s'éloignent plus ou moins de la réalité ; reste que l'élément tétraèdre linéaire T4 ne peut converger que seulement après un plus grand nombre d'éléments de division

A partir du graphe on peut tirer les conclusions suivantes

- Les solutions numériques se rapprochent des solutions exactes lorsqu'on utilise des éléments quadratiques que se soit en 2D ou en 3D
- Les hypothèses simplificatrices théoriques surdimensionnent toujours la réalité.

Etude 02 :

L'étude comparative a été faite en vue de voir l'influence du raffinement du maillage sur les résultats et cela dans le domaine 2D pour l'éléments Q 8, ainsi Soit la poutre suivante



$E = 32\,000\,000,00 \text{ kN/m}^2$
 $\nu = 0,21$
 $b = 0,40 \text{ m}$
 $L = 4,00 \text{ m}$
 $h = 0,50 \text{ m}$

	Nb Nœuds	Nb Eléments
P_16	65	16
P_64	233	64
P_128	457	128
P_256	905	256
P_512	1 801	512

Les résultats donnés par la RDM le long de la poutre sont donnés dans le tableau suivant

.x	y	w(x,y)
0.25	0	-0.016
0.25	0.25	-0.0145
0.25	0.5	-0.01302
0.25	0.75	-0.01156
0.25	1	-0.01013
0.25	1.25	-0.00875
0.25	1.5	-0.00743
0.25	1.75	-0.00618
0.25	2	-0.00501

.x	y	w(x,y)
0.25	2.25	-0.003935
0.25	2.5	-0.002965
0.25	2.75	-0.002113
0.25	3	-0.00139
0.25	3.25	-0.000807
0.25	3.5	-0.000377
0.25	3.75	-0.00011
0.25	4	-2e-05

Tableau 14 Résultats RDM

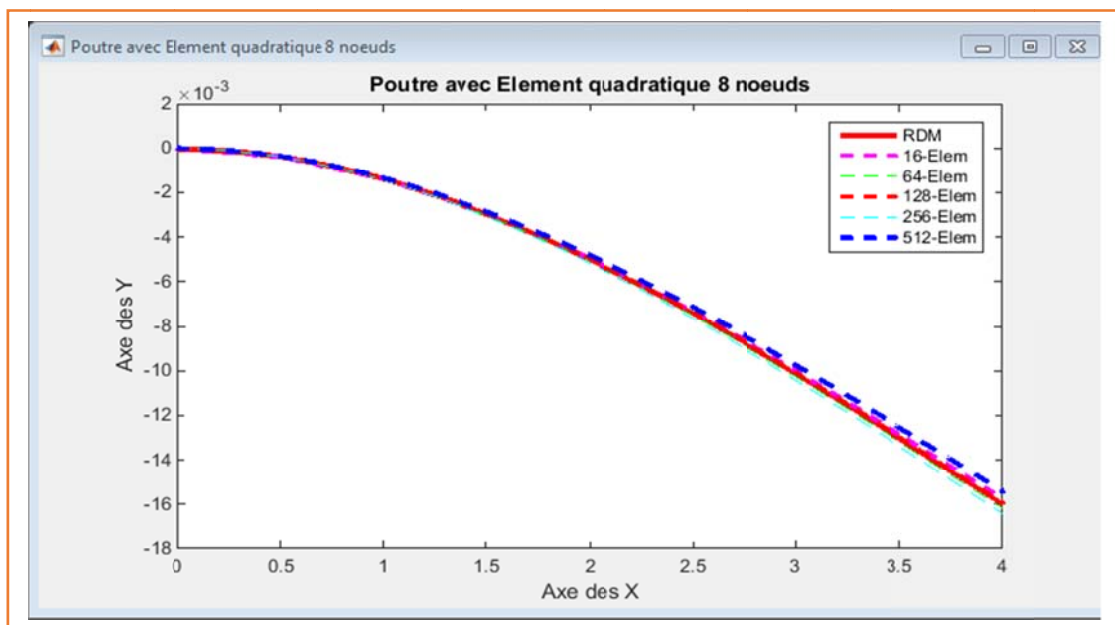


Figure 142 : évolution du déplacement le long de la poutre pour différents nombre de division par d'élément Q8

Etude 03 :
 domaine 2D Elément triangle Quadratique T6

	Nb Nœuds	Nb Eléments
P_32	81	32
P_64	153	64
P_128	297	128
P_256	585	256
P_512	1 161	512
P_1024	2 313	1024

Tableau 15 ; division élément T6

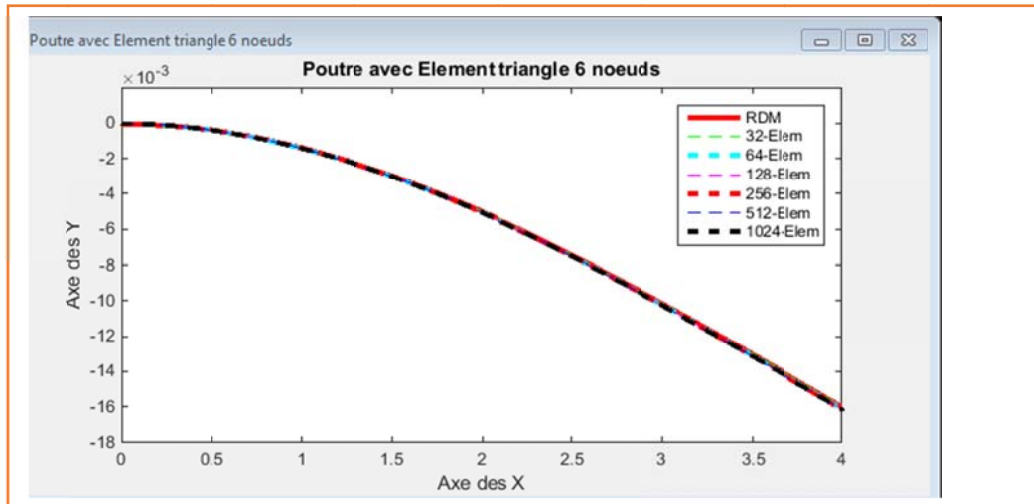


Figure 143 : évolution du déplacement le long de la poutre pour différents nombre de division par d'élément T6

Etude 04 :
 Domaine 3D _ Elément tétraèdre linéaire T4

	Nb Nœuds	Nb Eléments
P_24	18	24
P_48	27	48
P_96	45	96
P_192	81	192
P_384	153	384
P_768	297	768
P_1536	585	1536

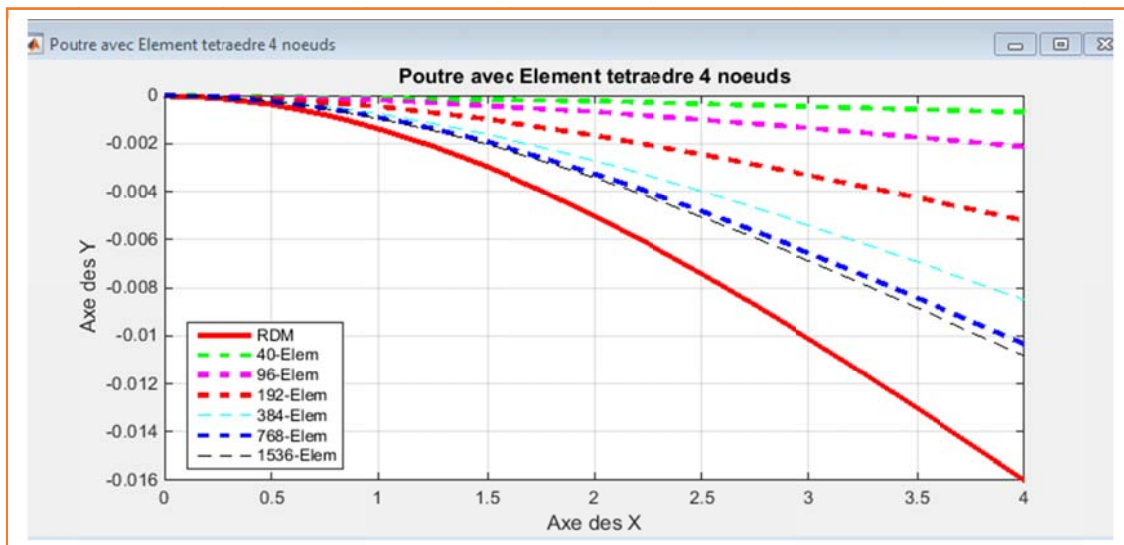


Figure 144 : évolution du déplacement le long de la poutre pour différents nombre de division par d'élément T4

Etude 05 :

Domaine 3D _ Elément cubique linéaire C8

	Nb Nœuds	Nb Eléments
P_4	18	4
P_8	27	8
P_16	45	16
P_32	81	32
P_64	153	64
P_128	297	128
P_256	585	256

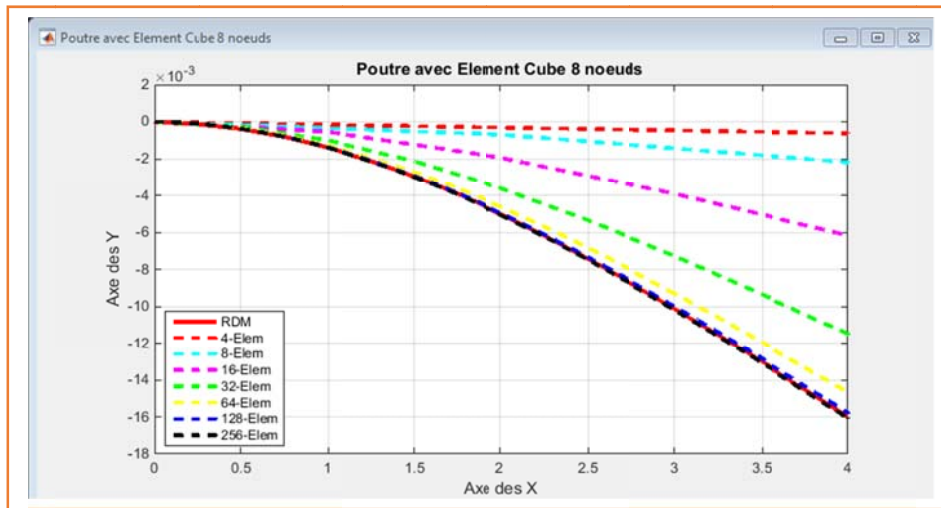


Figure 145 : évolution du déplacement le long de la poutre pour différents nombre de division par d'élément C8

Etude 06 :

Domaine 3D _ Elément hexaèdre C20

	Nb Nœuds	Nb Eléments
P_4	51	4
P_8	81	8
P_16	141	16
P_32	261	32
P_64	501	64
P_128	981	128
P_256	1 941	256

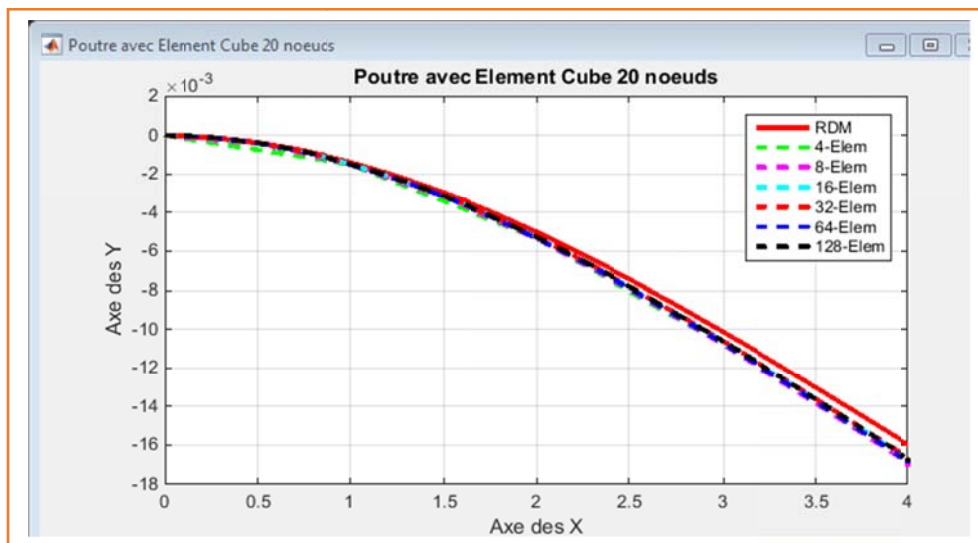


Figure 146 : évolution du déplacement le long de la poutre pour différents nombre de division par d'élément C 20

Chapitre 5

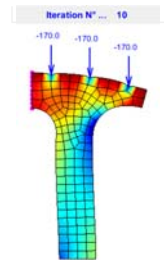
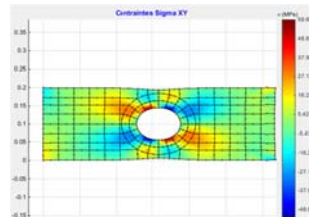
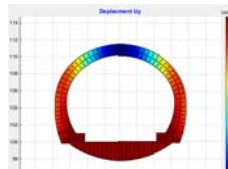
Calculs numériques

5.1 Introduction

Dans cette partie du projet on va présenter quelques exemples de calculs de différentes structures et problèmes réels que nous jugerons intéressants pour montrer les capacités du logiciel **MARS**.

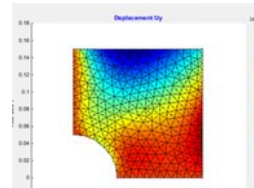
Elément fini Q8 :

- Pilier d'un ouvrage d'art.
- Plaque trouée.
- Tunnel.



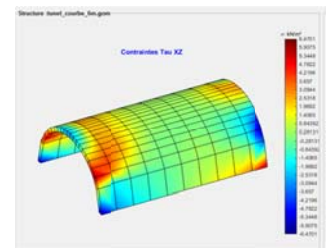
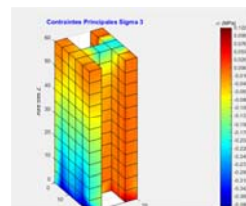
Elément fini T6 :

- Plaque métallique.



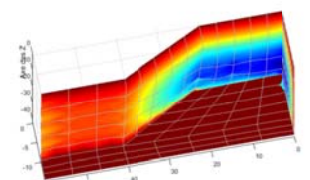
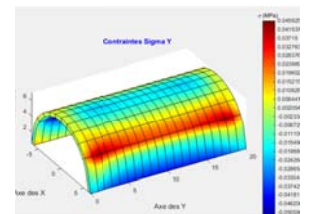
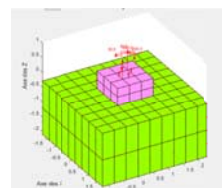
Elément fini cube C8 :

- Dallot 3D sous forme arc béton.
- Poteau métallique H (Acier).



Elément fini Hexa C20 :

- Dallot 3D sous forme arc (béton).
- Semelle isolée (béton,sol).
- Talus en terre (sol).



5.2 Apl design Q8

5.2.1 Exemple 01

Soit à analyser le pilier du pont illustré sur la figure 146 donné [Ref \[11\]](#) . Il est soumis à six charges concentrées de 170 kN chacune. Le matériau est du béton armé avec un module d'élasticité de 50000 MPa et un coefficient de Poisson de 0,17. Supposons que le support de base soit extrêmement rigide.

La première étape consiste à trouver des moyens de simplifier le modèle. L'épaisseur de la jetée est égale à 0,70 m, ce qui est relativement faible par rapport aux dimensions horizontale et verticale. Le pilier peut donc être analysé comme un problème de contrainte plane. De plus, le chargement et la géométrie sont symétriques, donc seule la moitié de la pile peut être analysée. La deuxième étape consiste à identifier les conditions aux limites. La troisième étape consiste à construire un maillage approprié du domaine. En effet, la qualité des résultats numériques dépend beaucoup de la qualité du maillage. Cependant, il s'agit probablement de la tâche la plus difficile et la plus longue de toute analyse par éléments finis, en particulier lorsque des géométries complexes sont prises en compte.

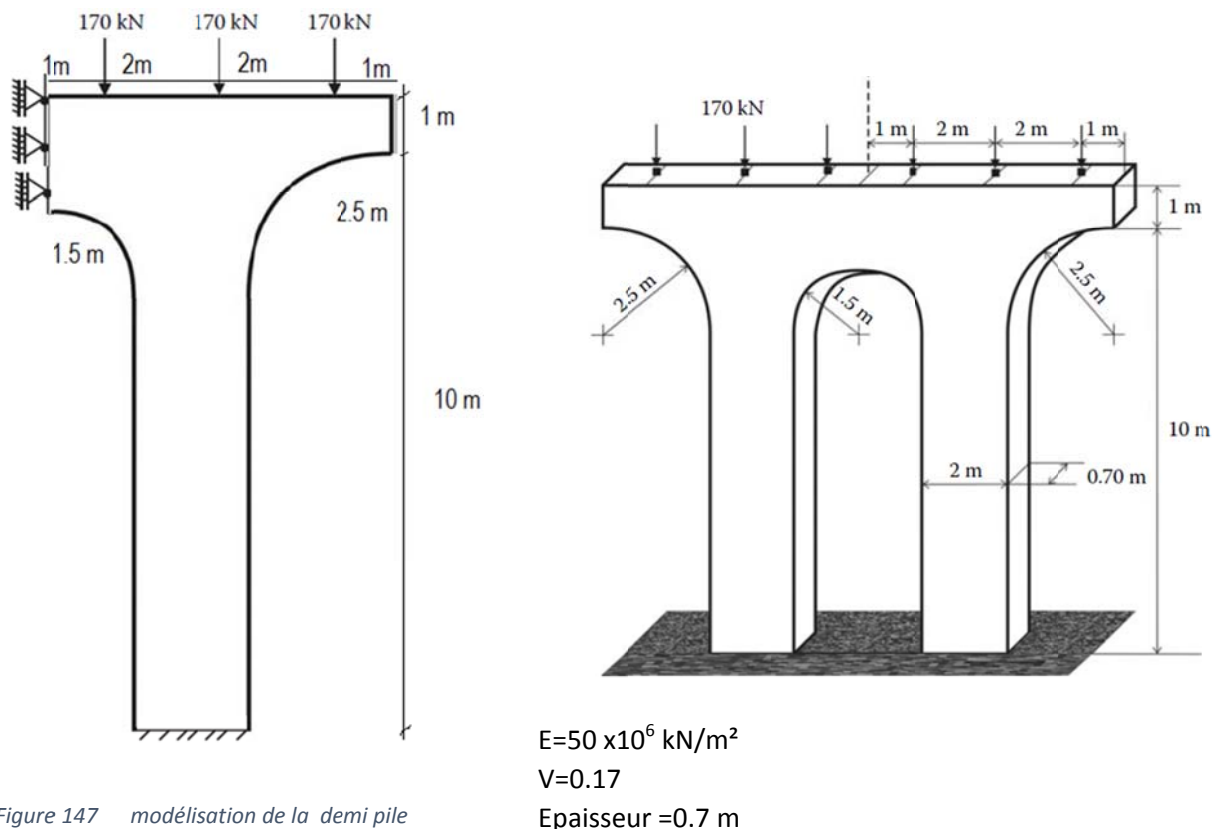


Figure 147 modélisation de la demi pile

Figure 148 : exemple 01 pile de pont

Vérification du maillage sous MARS pour détecter les anomalies probables

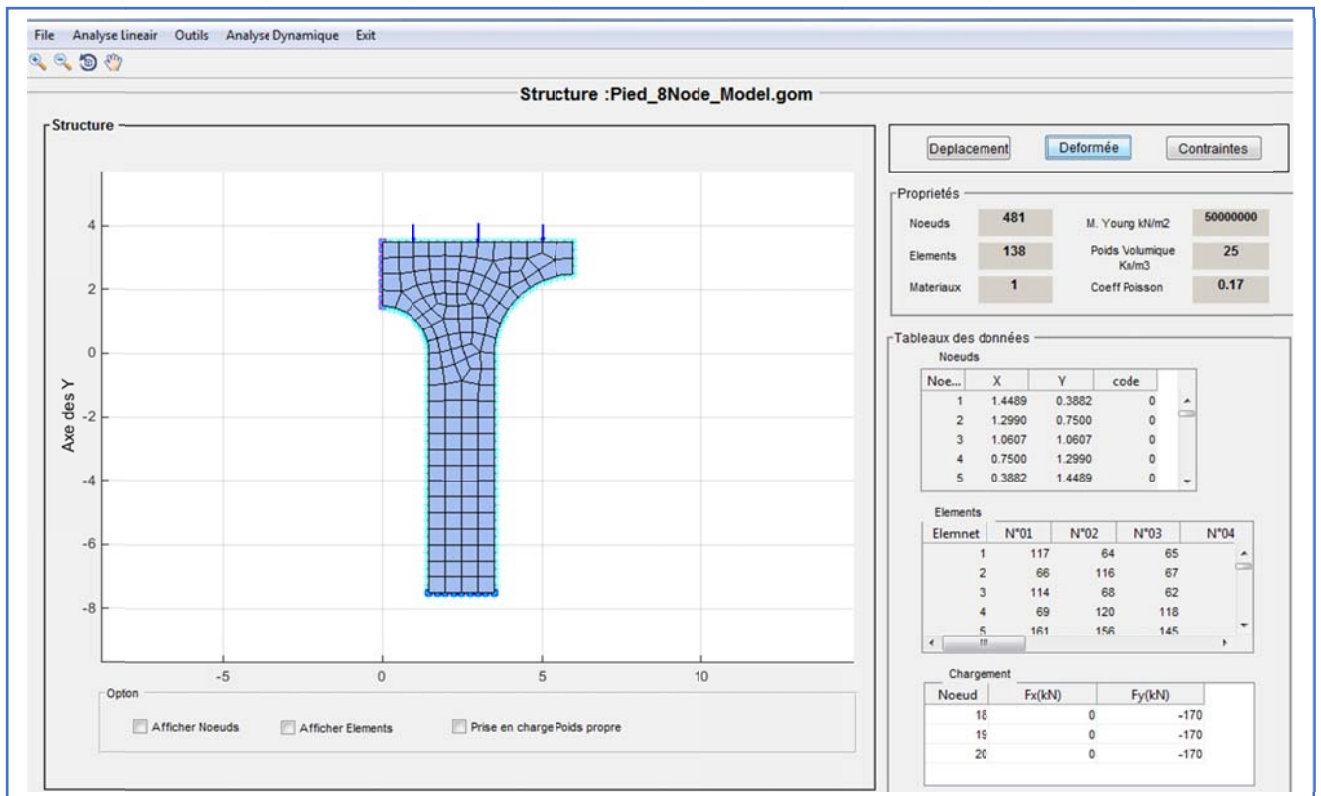


Figure 149 : affichage de la structure

Résultats après exécution du programme : déplacements aux nœuds ainsi que les contraintes aux points de Gauss et aux nœuds après interpolation.

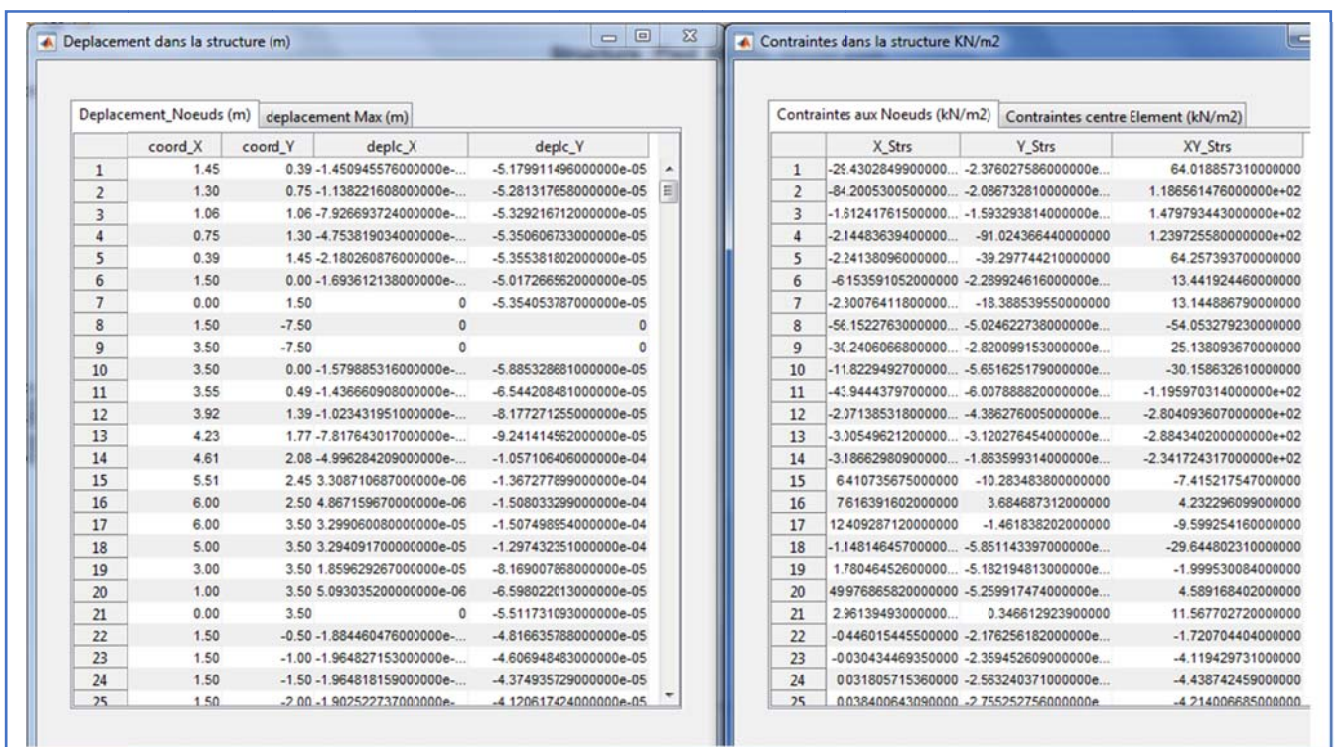
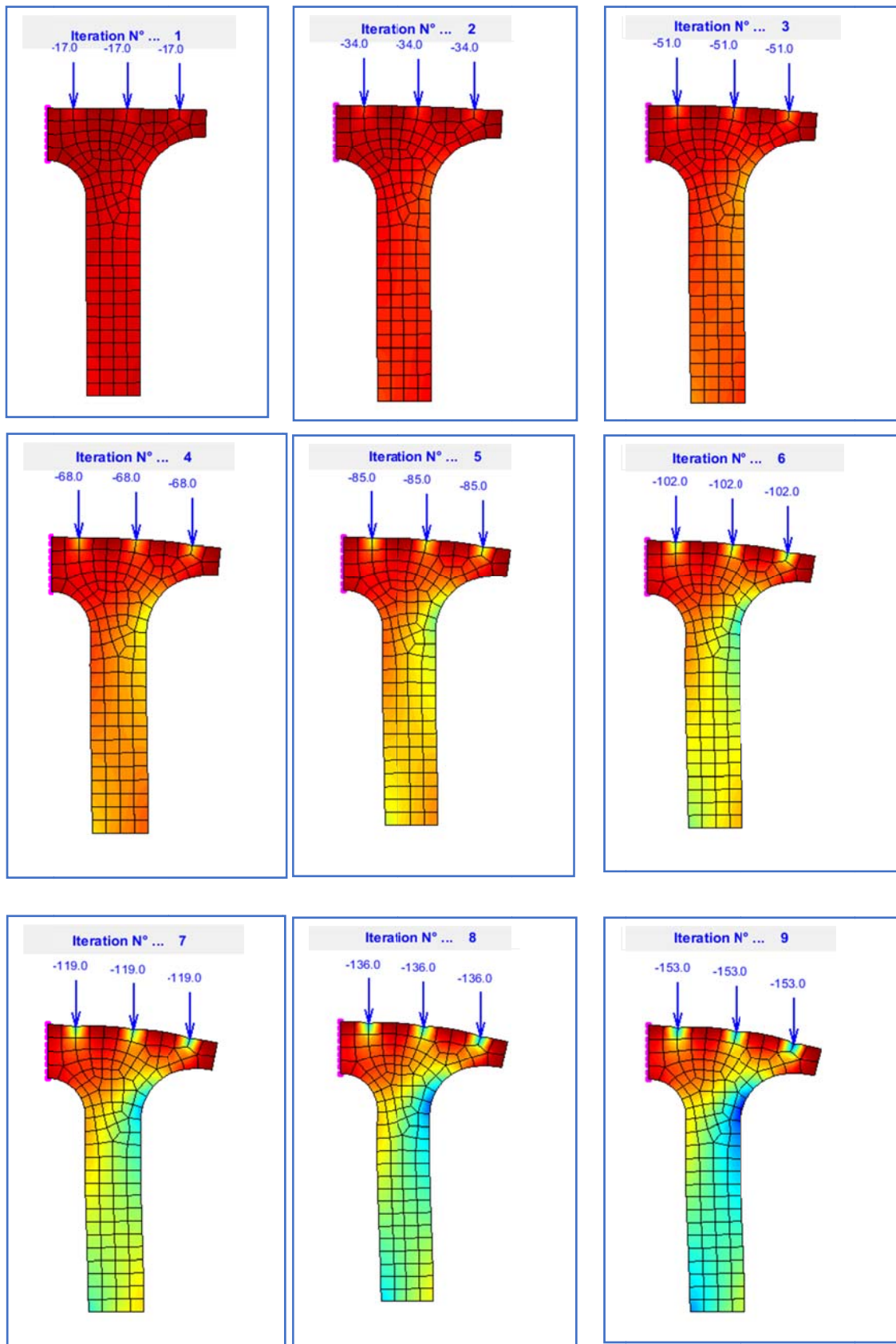
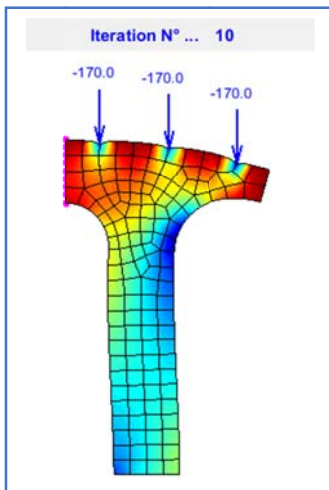


Figure 150 : résultats de déplacements et contraintes

ci-dessous une animation de la structure pour chaque itération du calcul ou il sera affiché les charges appliquées ainsi que la déformée correspondante.





On présente sur les figures suivantes les diagrammes des déplacements u_x et u_y ainsi que le diagramme des contraintes σ_x , σ_y et σ_{xy} et contraintes principales σ_1 et σ_2 .

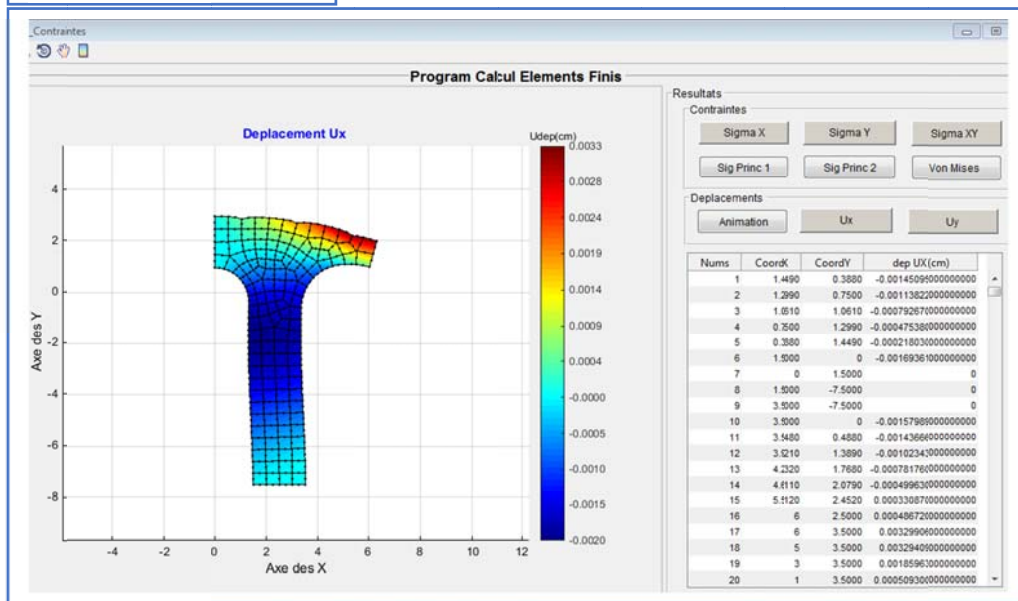


Figure 151 : déformée de la structure U_x avec un facteur d'échelle

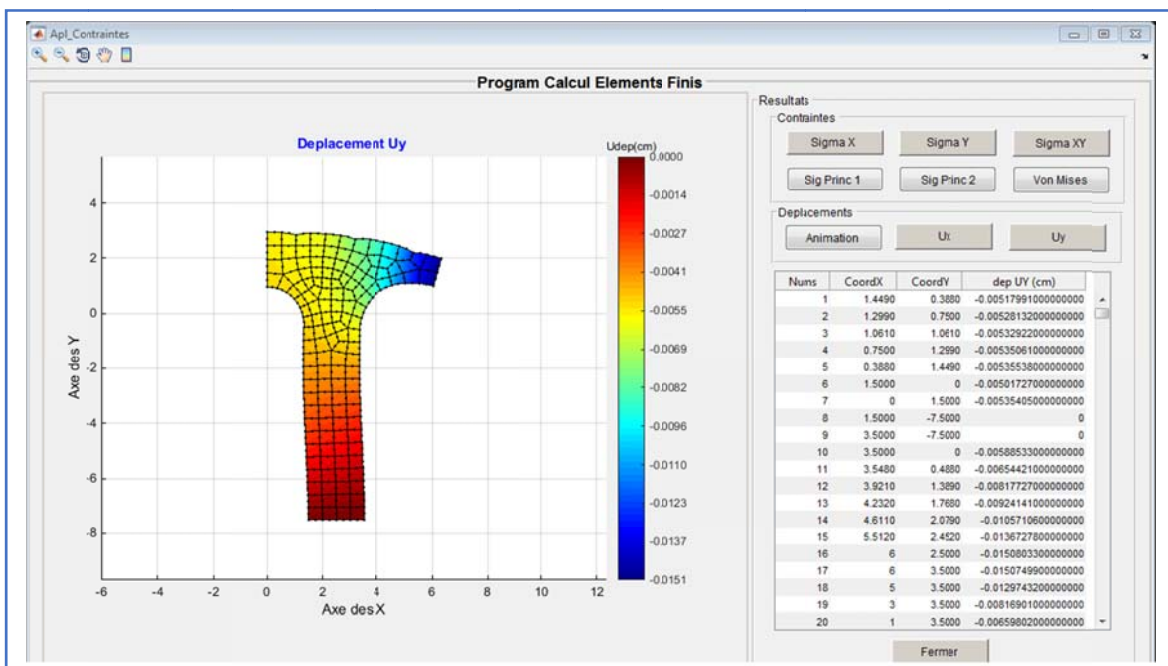


Figure 152 : déplacement U_y avec un facteur d'échelle

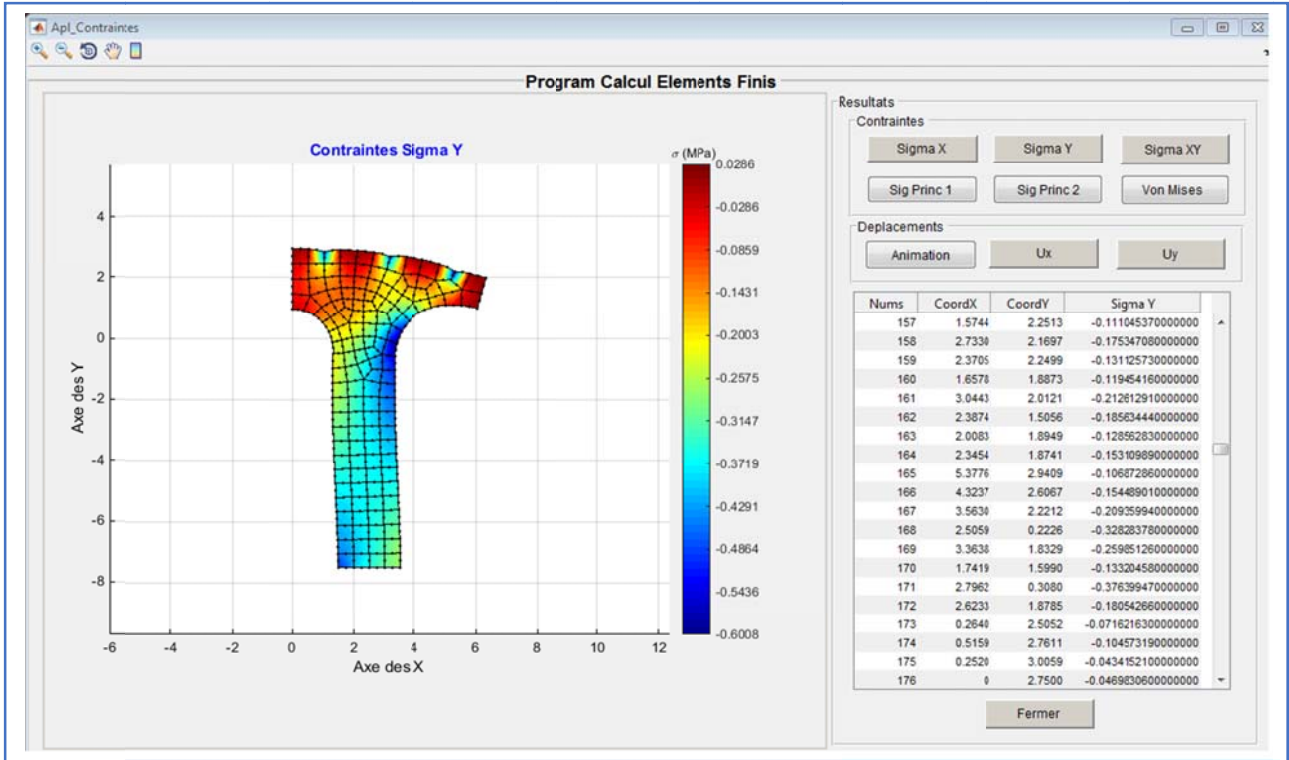


Figure 153 : distribution des contraintes σ_y

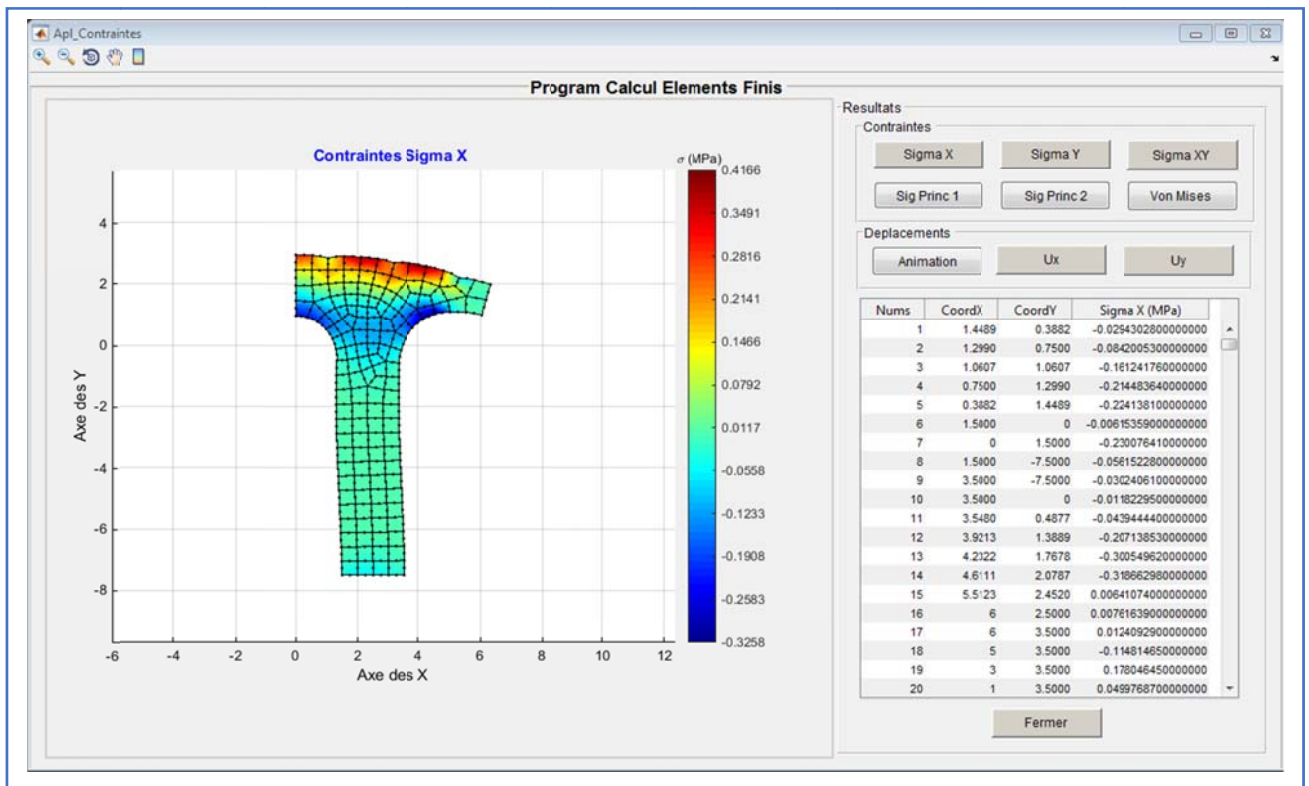


Figure 154 : distribution des contraintes σ_x

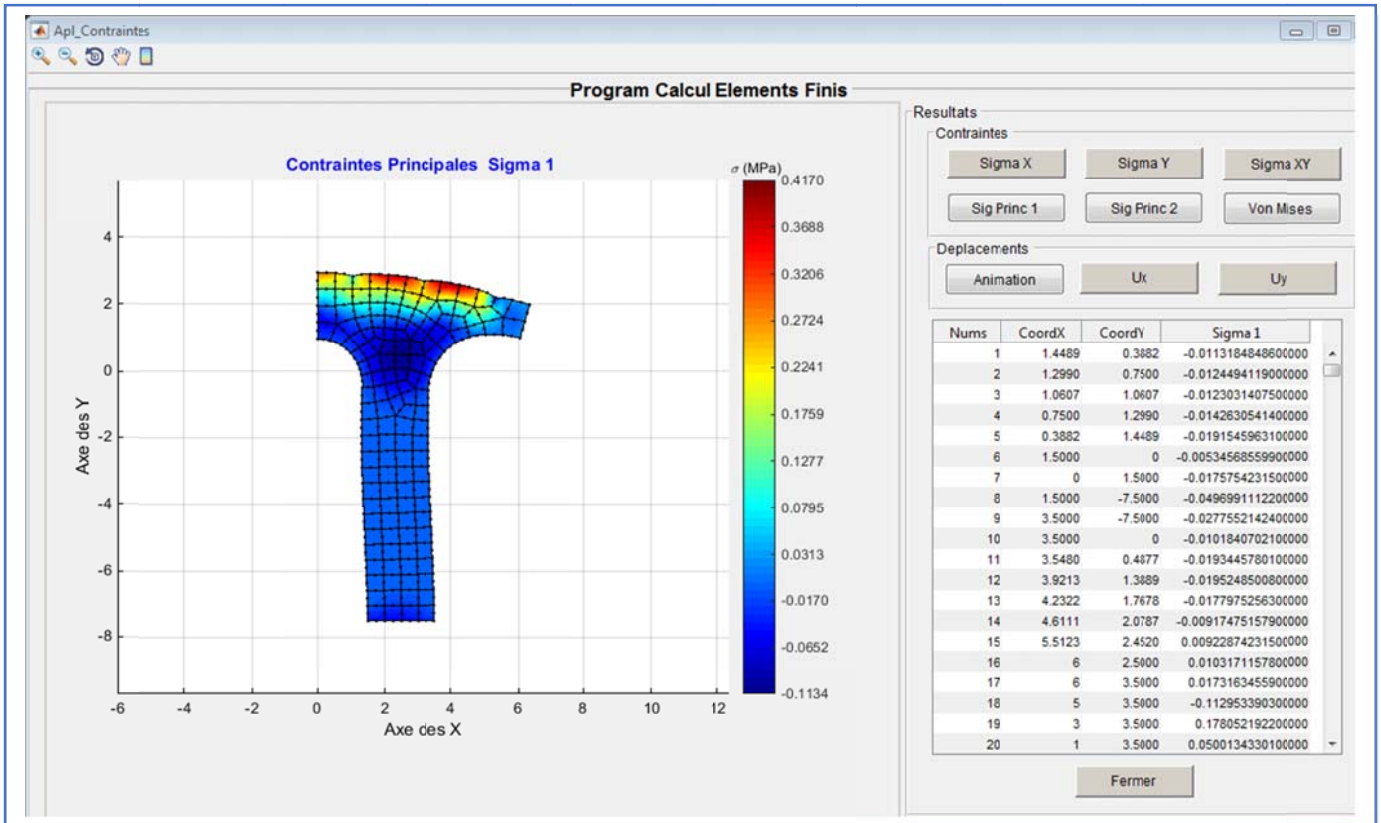


Figure 155 : distribution des contraintes σ principale 1

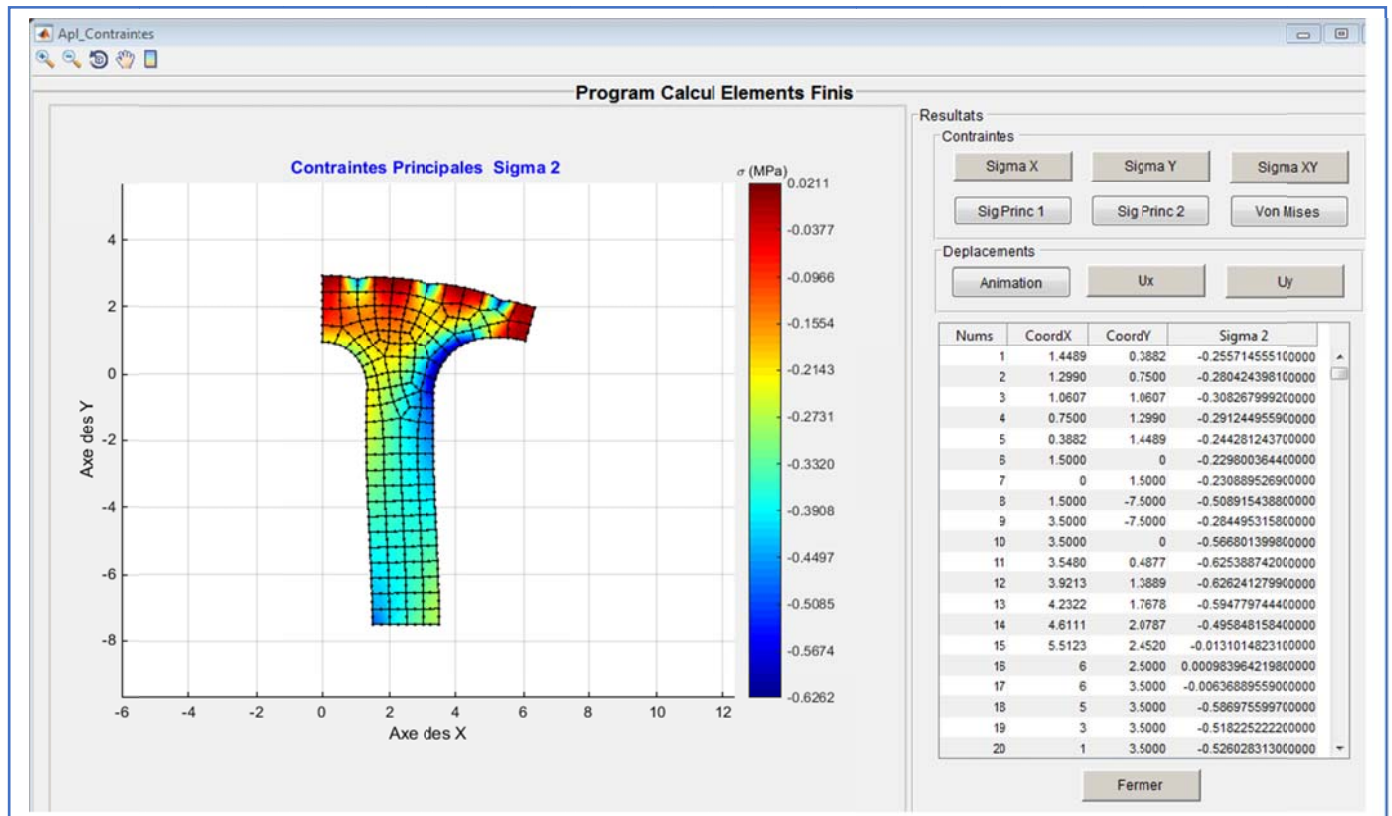


Figure 156 : distribution des contraintes σ principale 2

5.2.2 Exemple 02

Description :

Le deuxième exemple est une plaque métallique trouée d'une longueur de 60 cm est une largeur de 20 cm et une épaisseur de 7 mm, avec un trou au de rayon 5 cm.

Encastrée a son extrémité gauche et étirée par une force uniformément répartie à son extrémité droite.

Le maillage est réalisé avec 144 éléments Q8 et un nombre de 500 nœuds

Donc un problème de contrainte plane avec les caractéristiques suivantes : $E=210 \times 10^6 \text{ kN/m}^2$, $\nu=0.17$, $\rho=77 \text{ KN/m}^3$

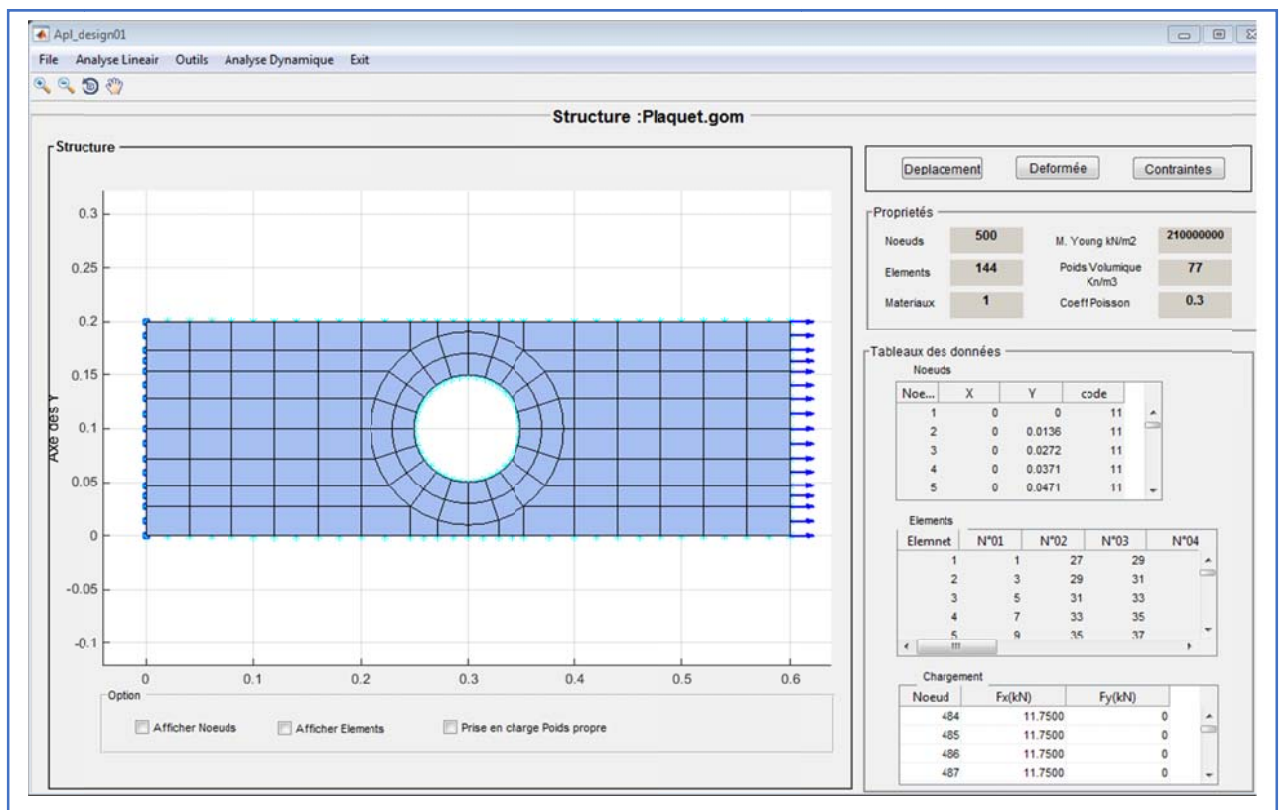


Figure 157 : maillage structure plaque trouée

Déformée de la structure avec un facteur d'échelle de 100.

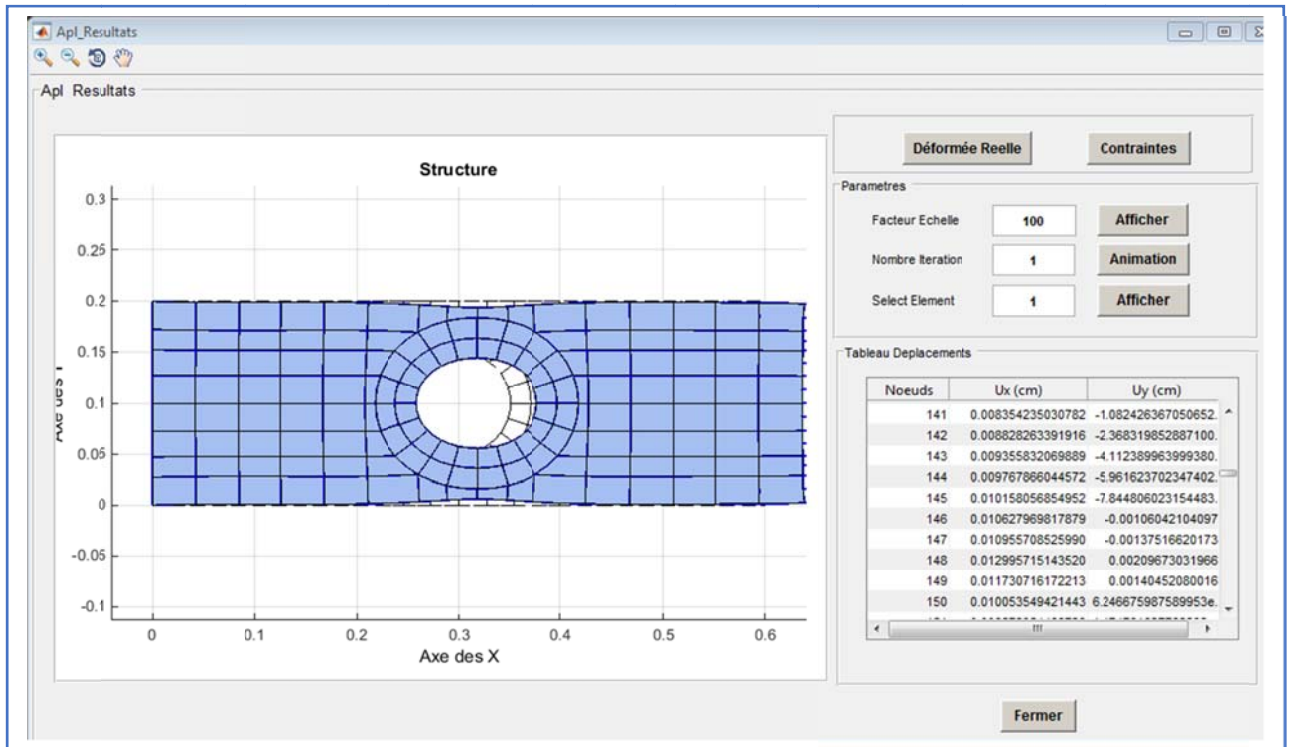


Figure 158 : déformée de la structure avec un facteur d'échelle

Les tableaux suivants donnent les déplacements aux nœuds, les déplacements maximums, les contraintes au nœuds et au centre des éléments finis.

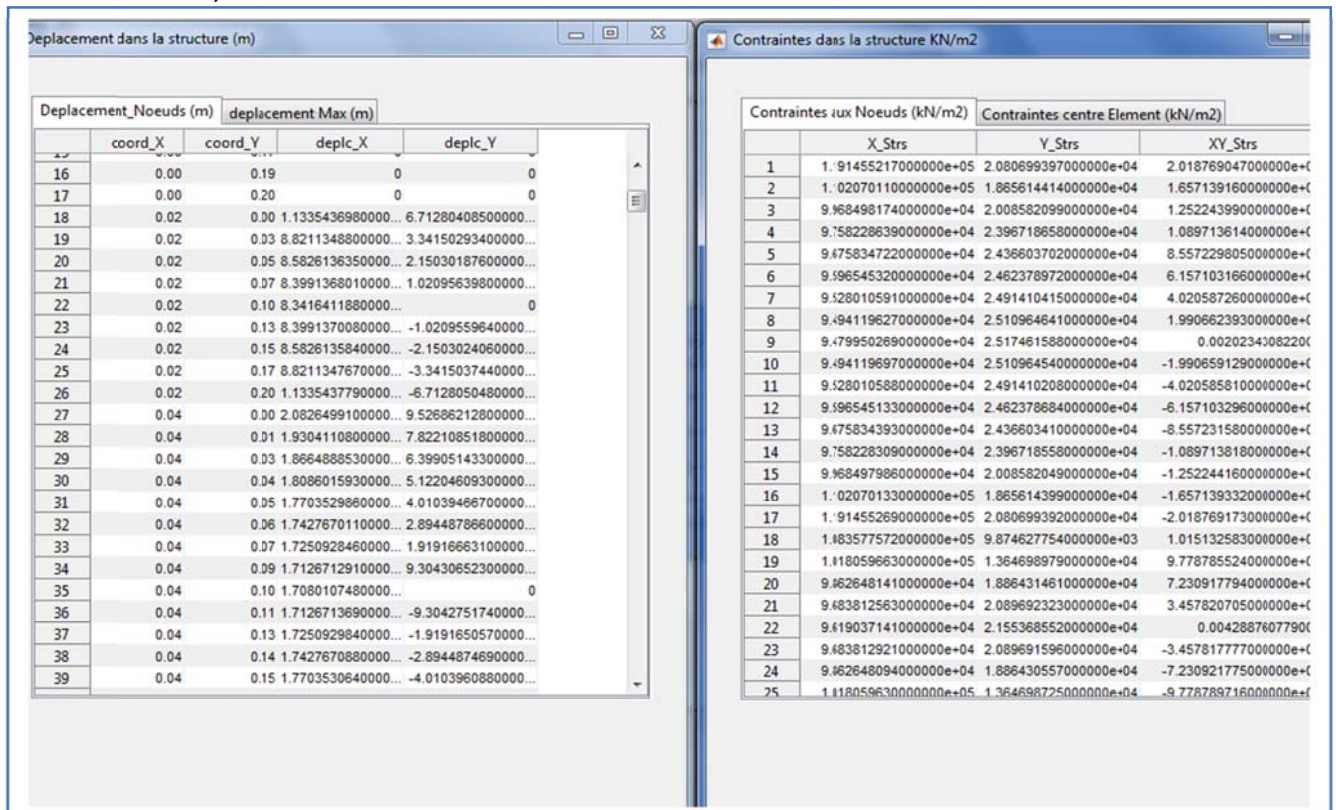


Figure 159 : résultats des déplacements et des contraintes

On présente sur les figures suivantes : Les diagrammes des contraintes σ_x , σ_y et σ_{xy} et contraintes principales σ_1 et σ_2 .

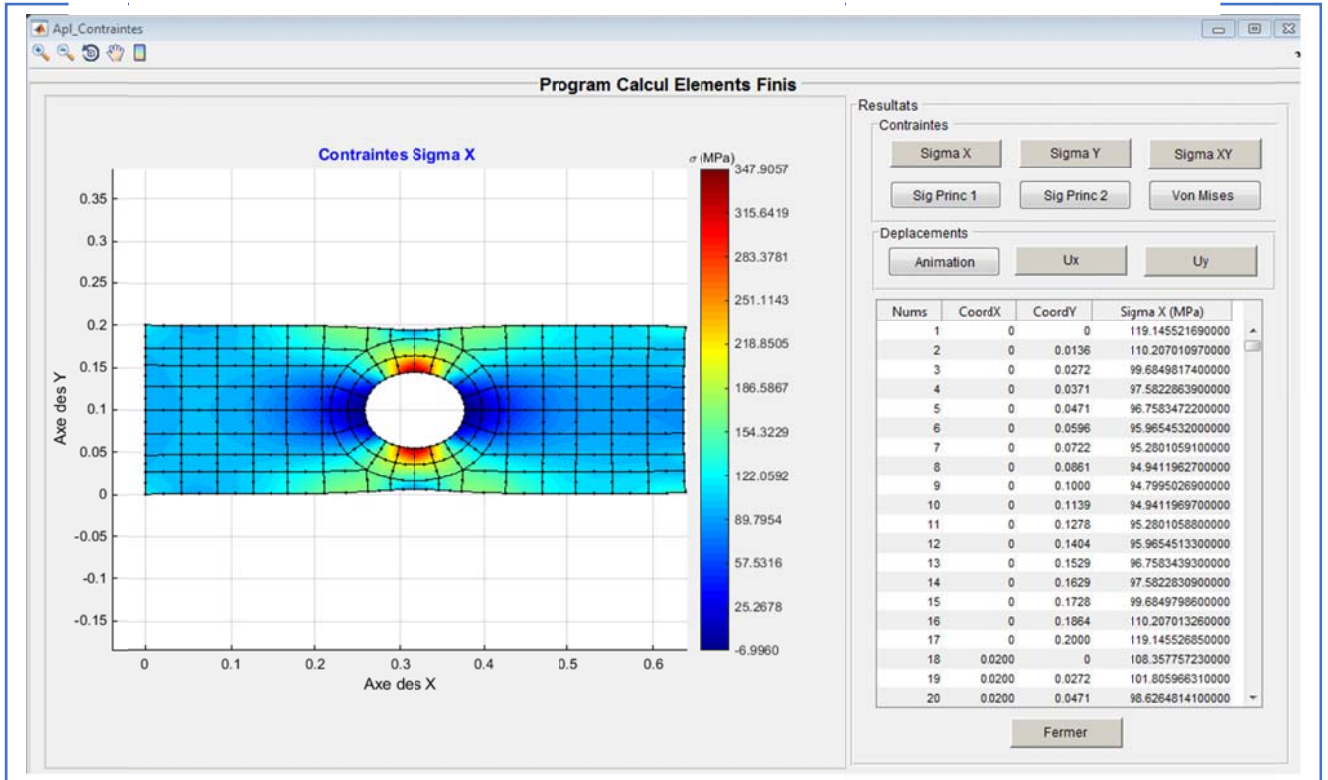


Figure 160 : distribution des contraintes σ_x

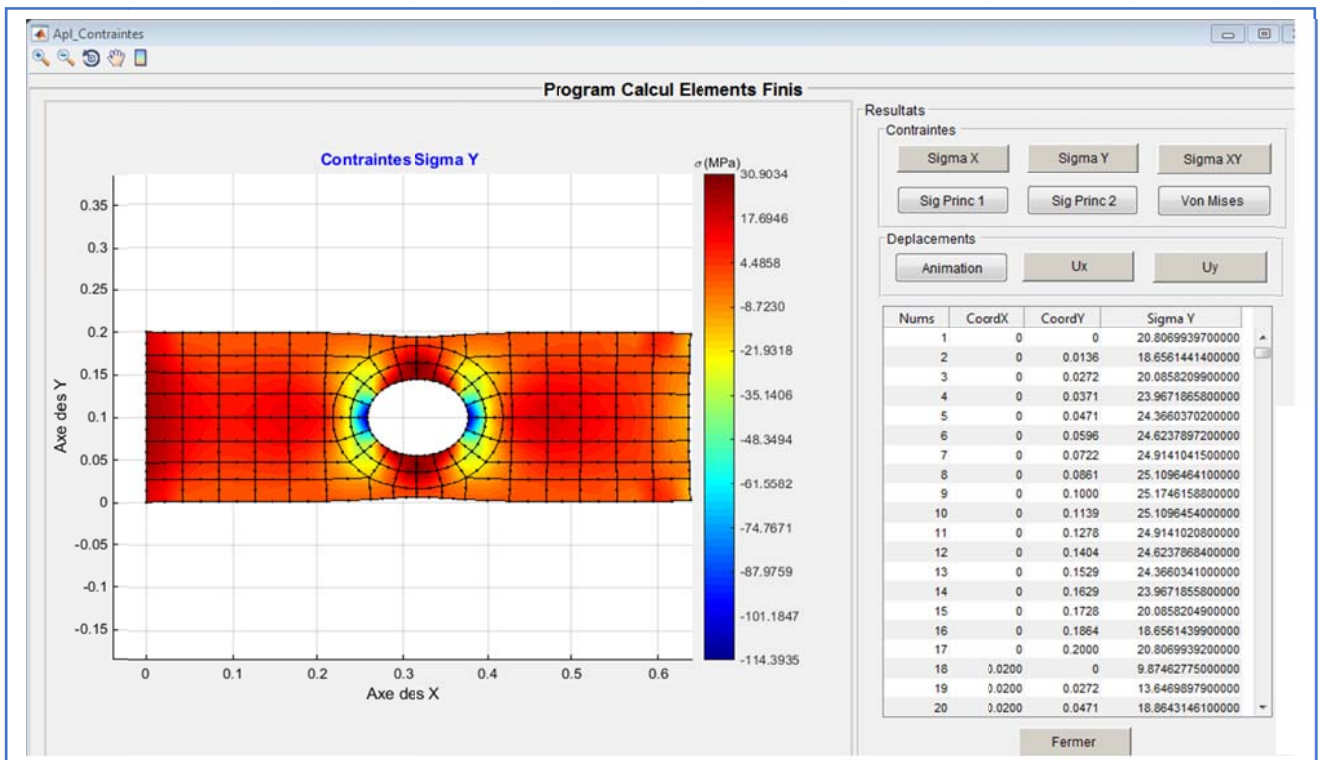


Figure 161 : distribution des contraintes σ_y

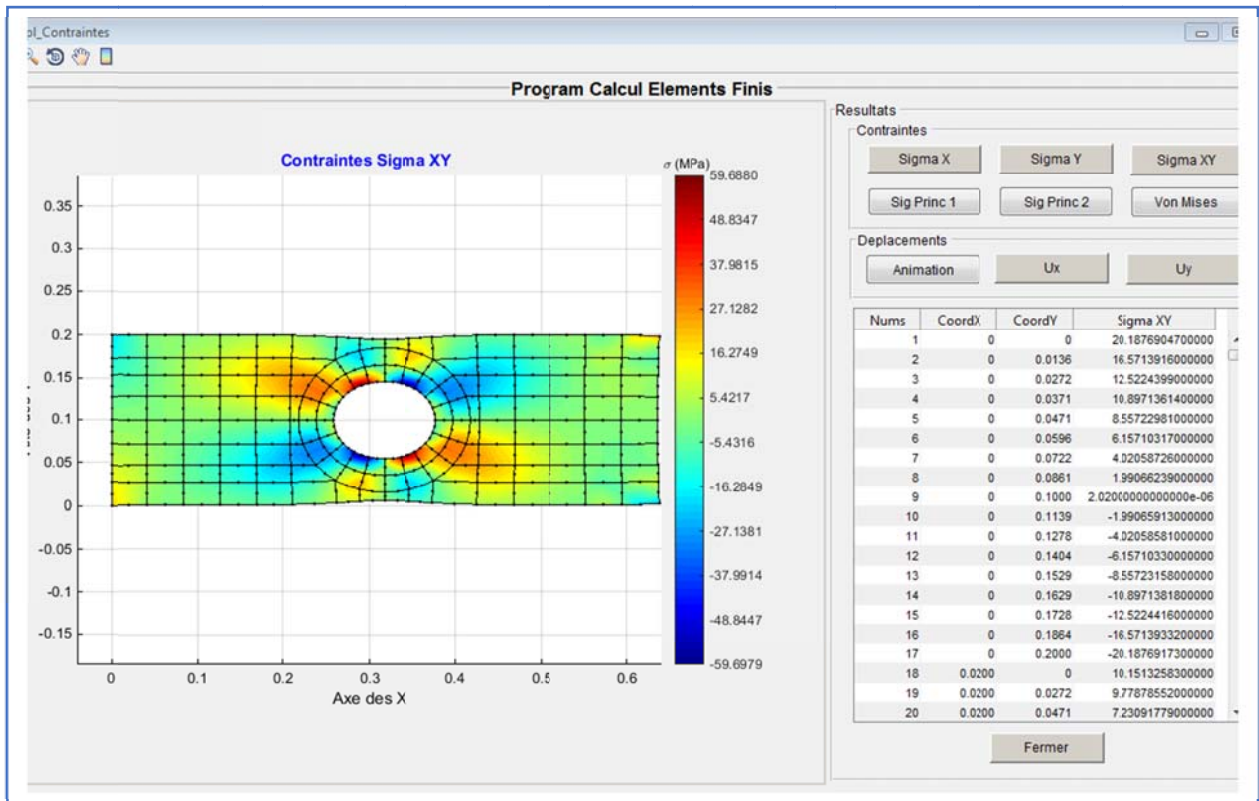


Figure 162 : distribution des contraintes σ_{xy}

5.2.3 Exemple 03

Description :

Le troisième exemple est un tunnel en béton armé, les dimensions sont données comme suit. dans cet exemple le tunnel est soumis a son poids propre seulement.

Encastrée a son extrémité a sa base.

Le maillage est réalisé avec 152 éléments Q8 et un nombre de 626 nœuds

Donc un problème de déformations

On présente sur la figure suivante le maillage du tunnel

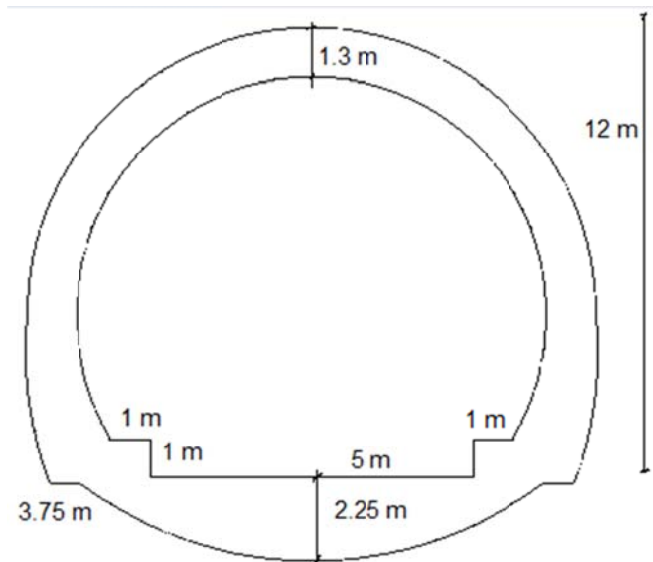


Figure 163 : structure tunnel

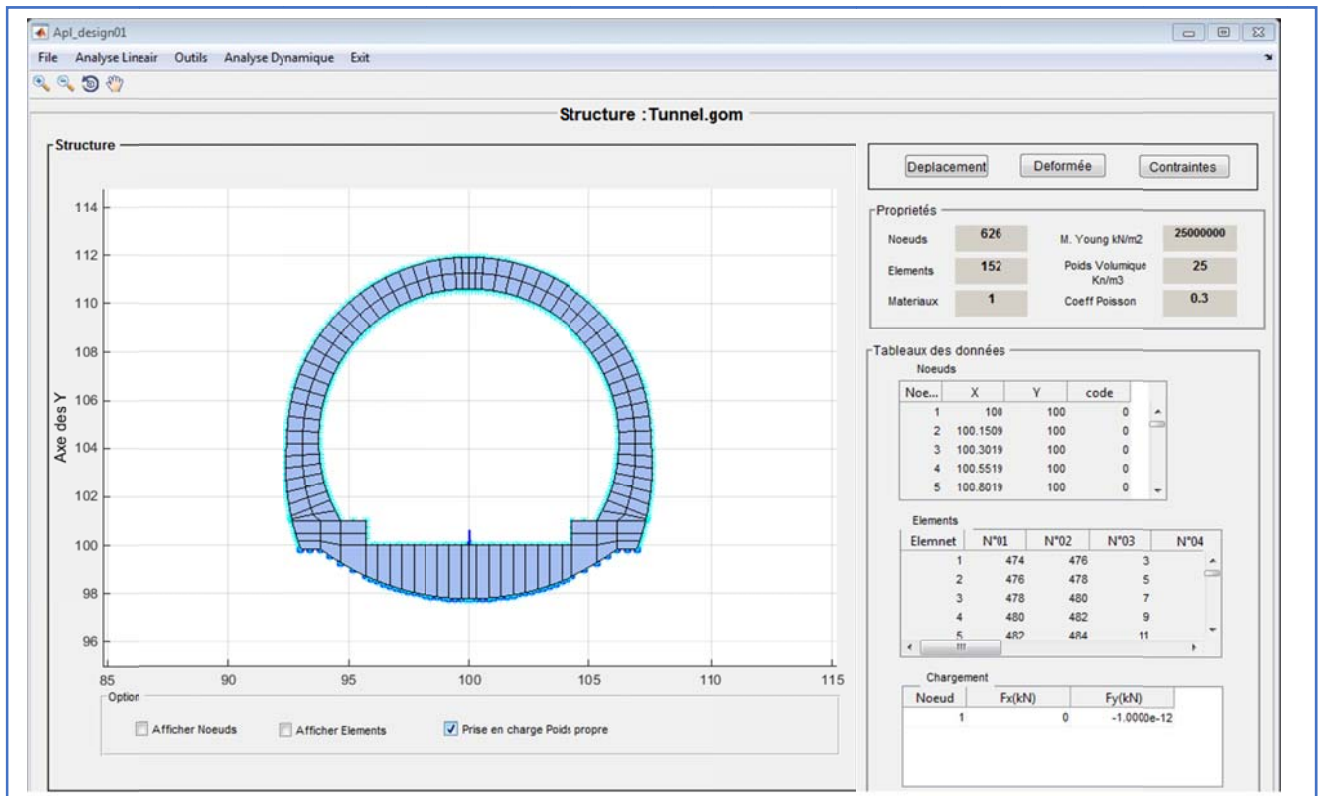


Figure 164 : maillage de la structure

Déformée du tunnel avec un facteur d'échelle de 1000.

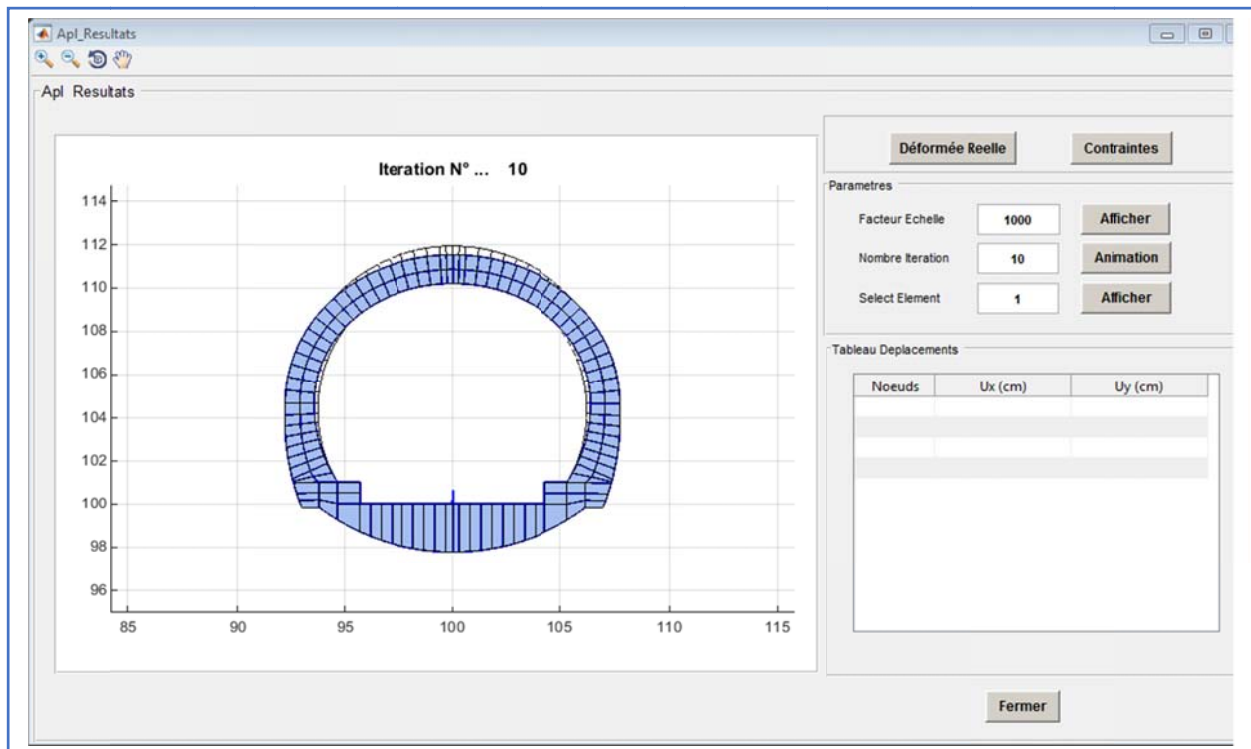


Figure 165 : déformée de la structure avec un facteur d'échelle

On présente sur les figures suivantes les diagrammes des déplacements u_x et u_y ainsi que le diagramme des contraintes σ_x , σ_y et σ_{xy} et contraintes principales ainsi que celle de von mises.

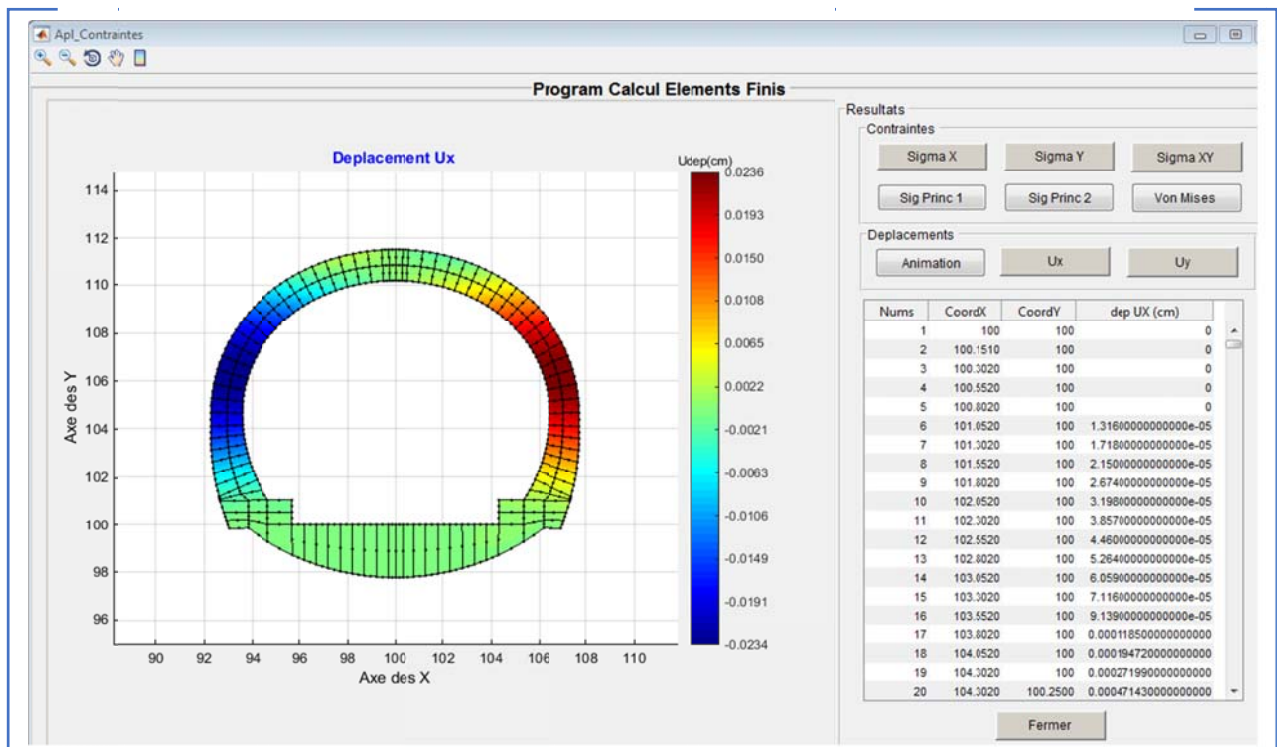


Figure 166 : déformée de la structure U_x

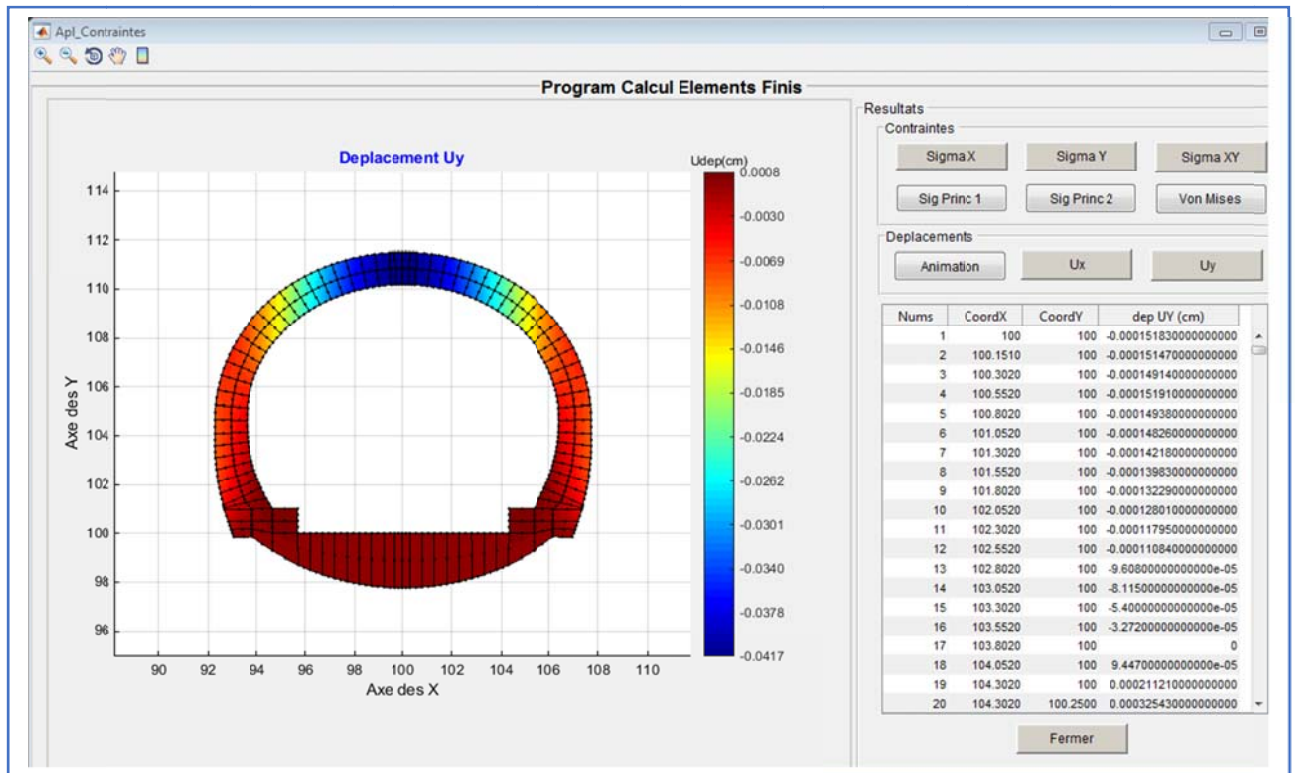


Figure 167 déformée de la structure Uy

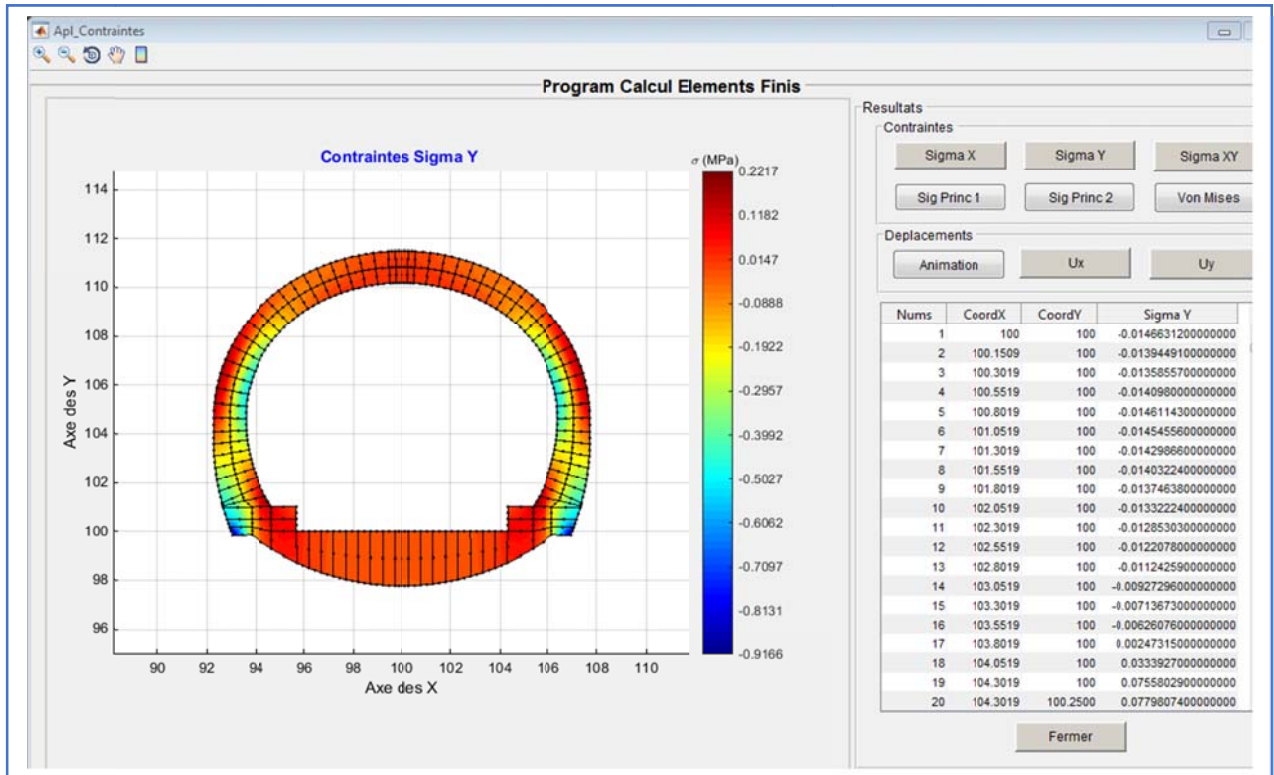


Figure 168 : distribution des contraintes σ_y

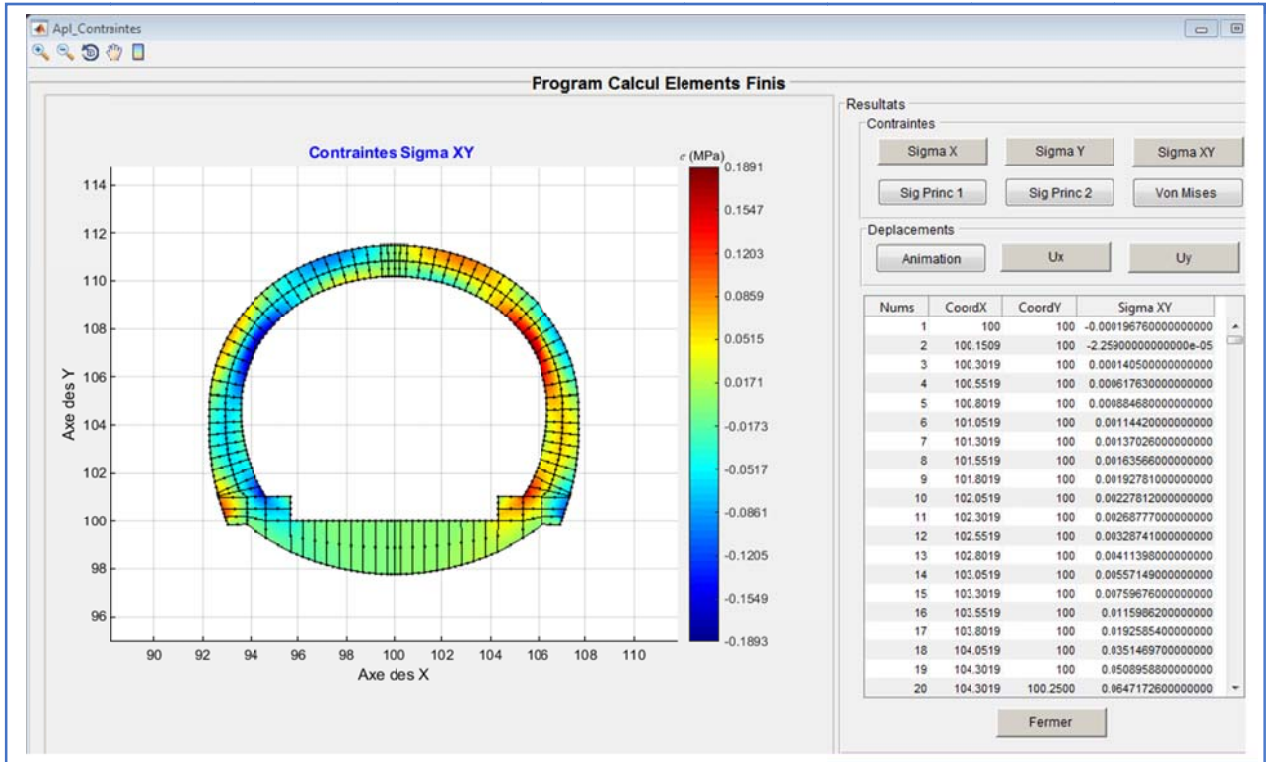


Figure 169 : distribution des contraintes σ_{xy}

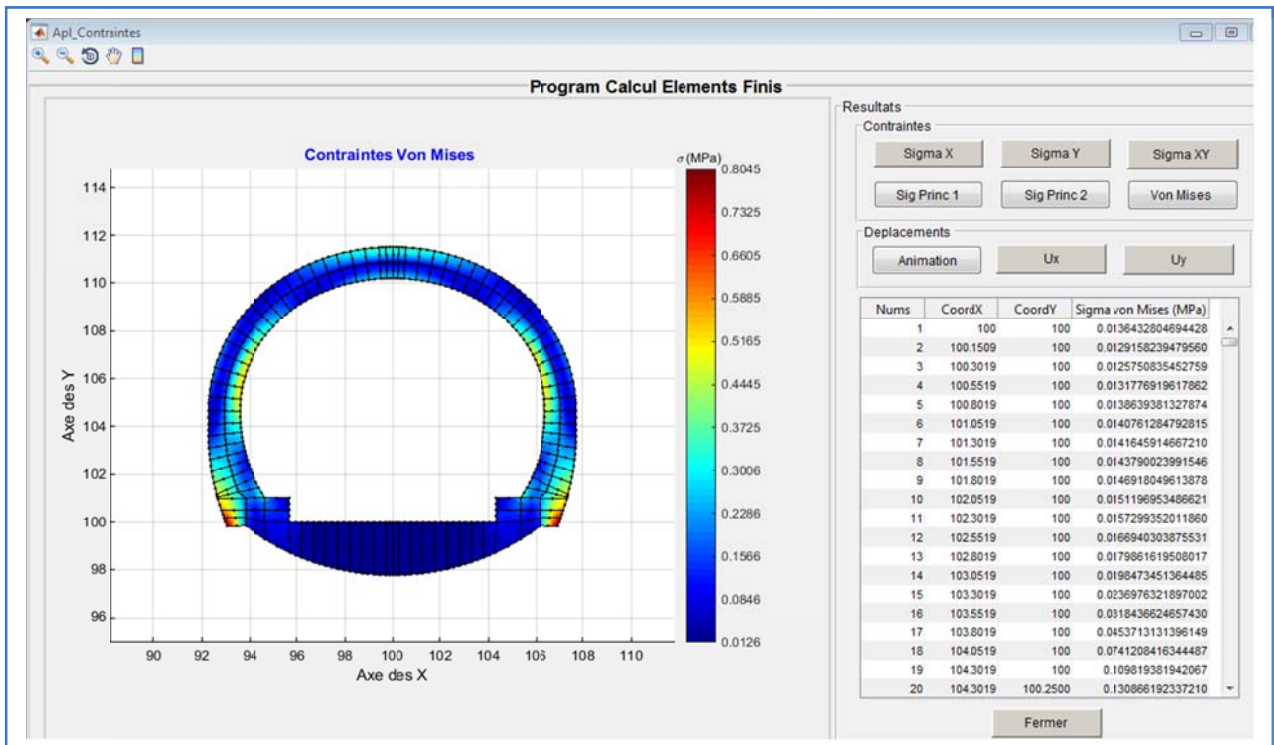


Figure 170 : distribution des contraintes de Von Mises

5.2.4 Exemple 04

Soit à étudier un terrain en talus comme l'indique la figure soumis a une charge $P = 600 \text{ kN}$ uniformément répartie sur 4 m

un rideau palplanche en acier avec les caractéristiques suivantes

sol
 $E = 6894.757$
 $\nu = 0.47$
 $\rho = 18$

rideau acier
 $E = 210 \text{ GPa}$
 $\nu = 0.3$
 $\rho = 78$

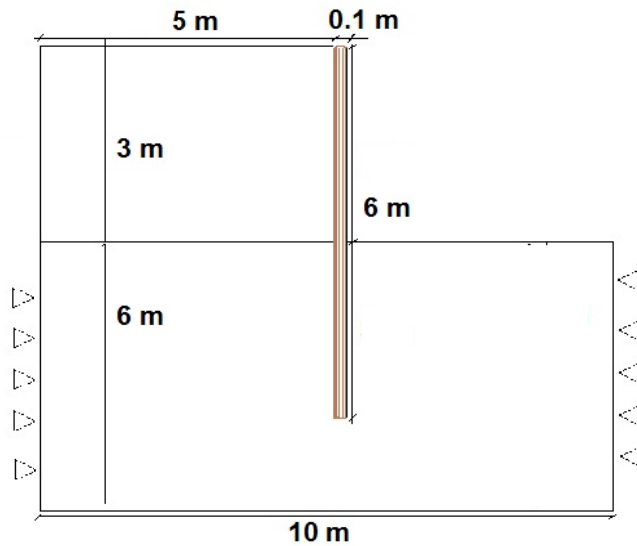


Figure 171 exemple rideau palplanche

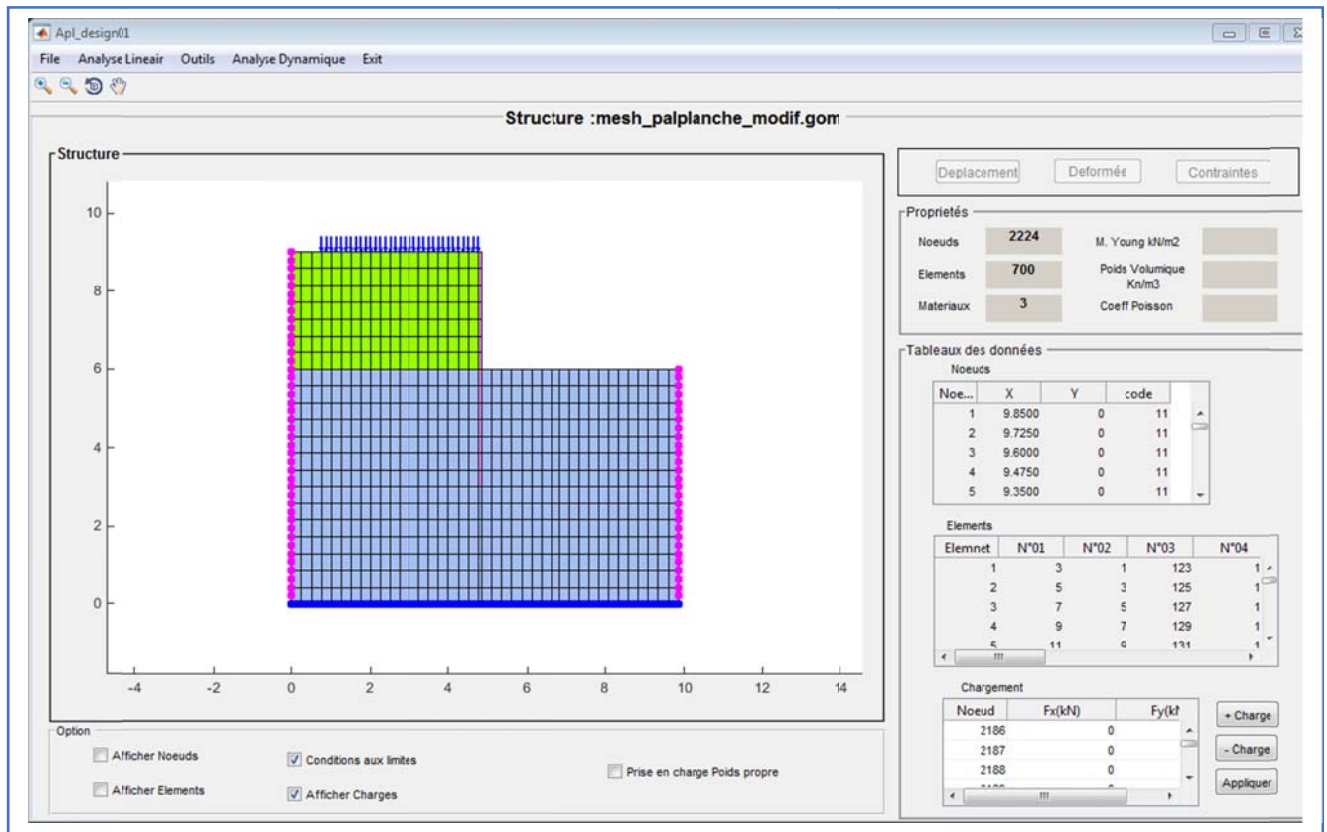


Figure 172 : maillage de l'exemple à étudier

Cas sans Palplanche

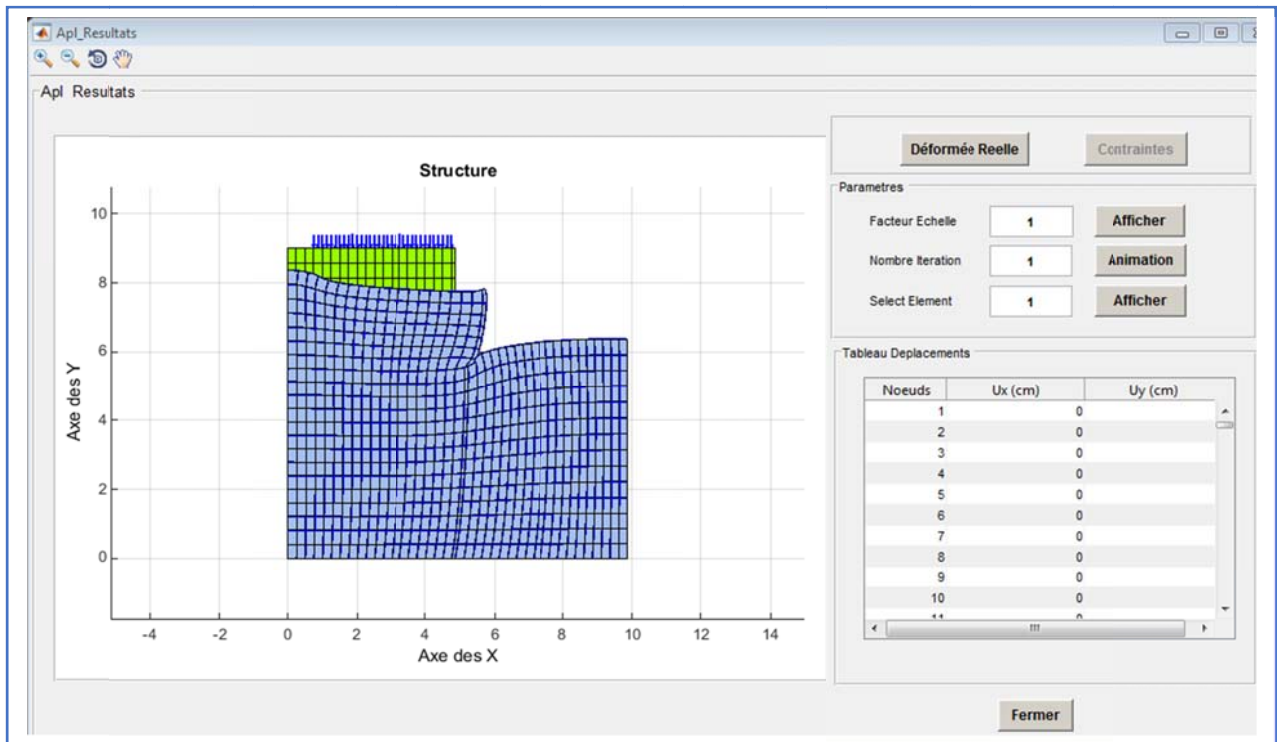


Figure 173 : déformée du sol sans Palplanche

Cas avec Palplanche

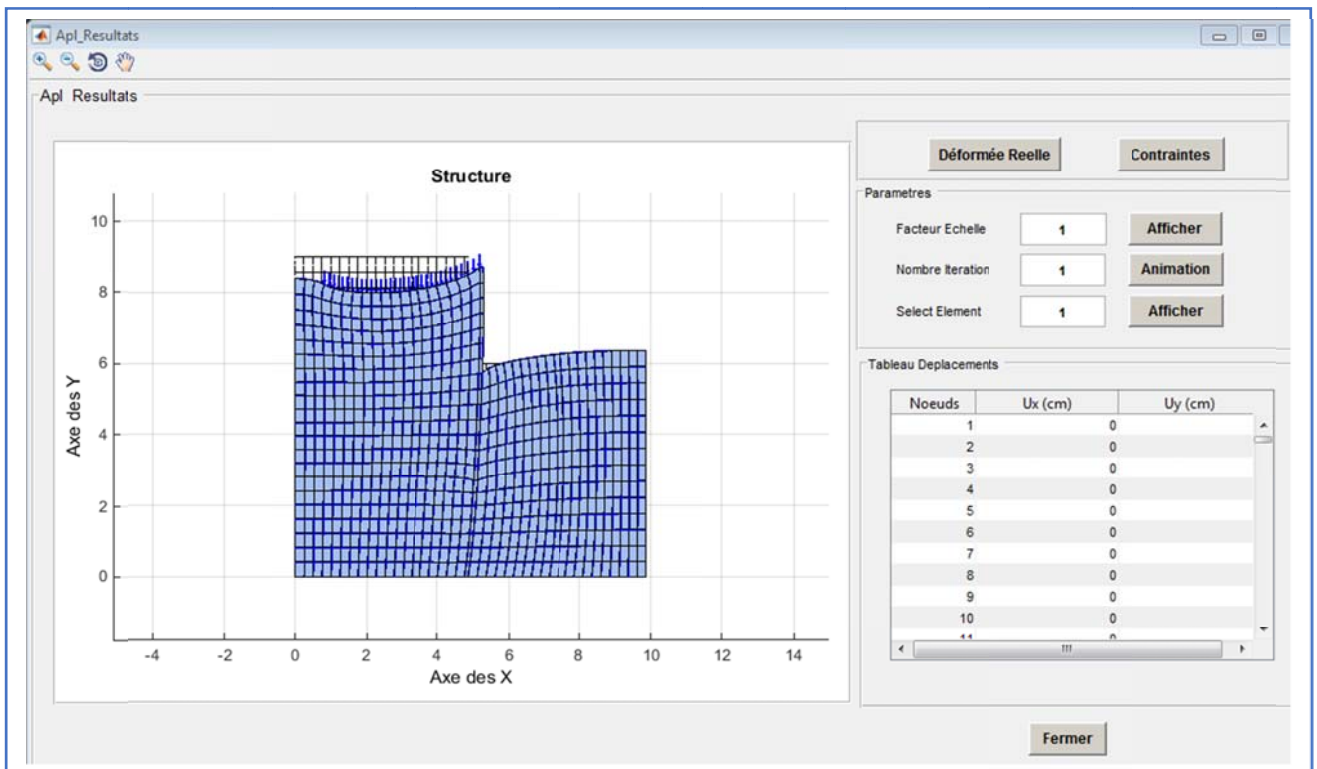


Figure 174 : déformée du sol sans Palplanche

Cas sans Palplanche

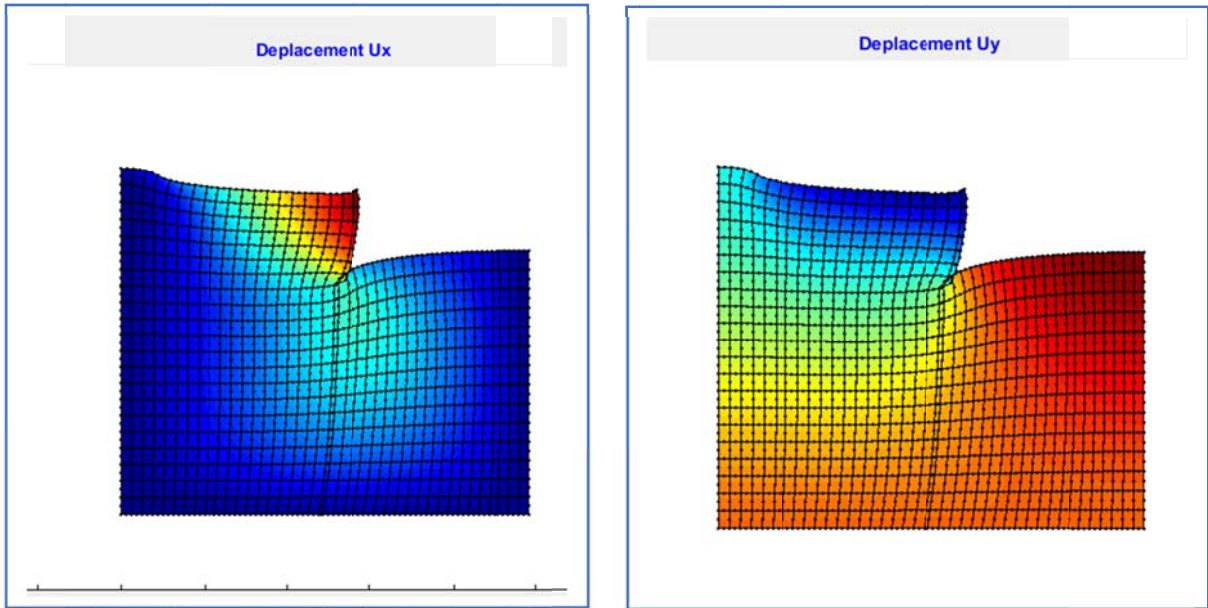


Figure 175 : déplacements et déformée du sol sans Palplanche

Cas avec Palplanche

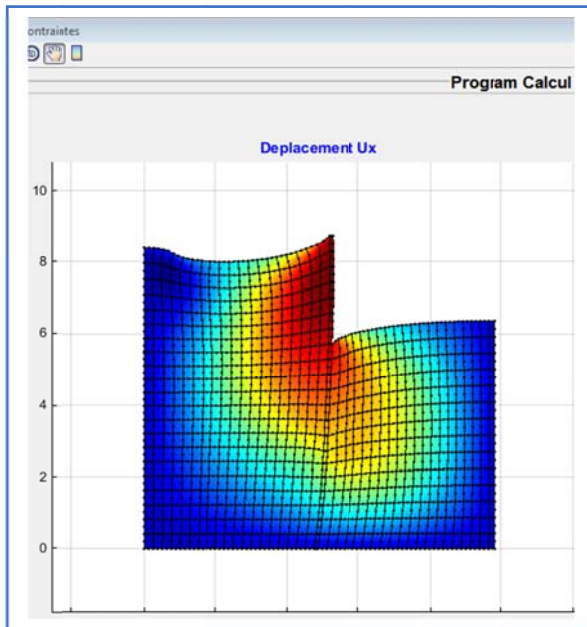


Figure 176 : déplacements et déformée du sol avec Palplanche

Cas sans Palplanche

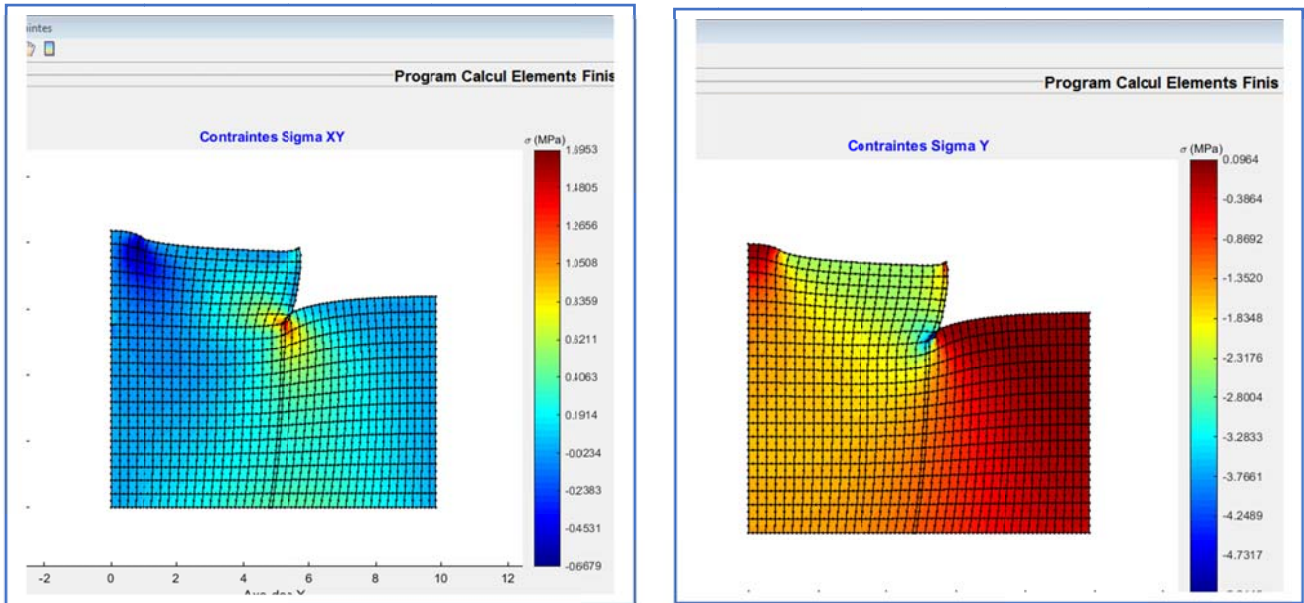


Figure 177 : distribution des contraintes σ_{xy} σ_y sans Palplanche

Cas avec Palplanche

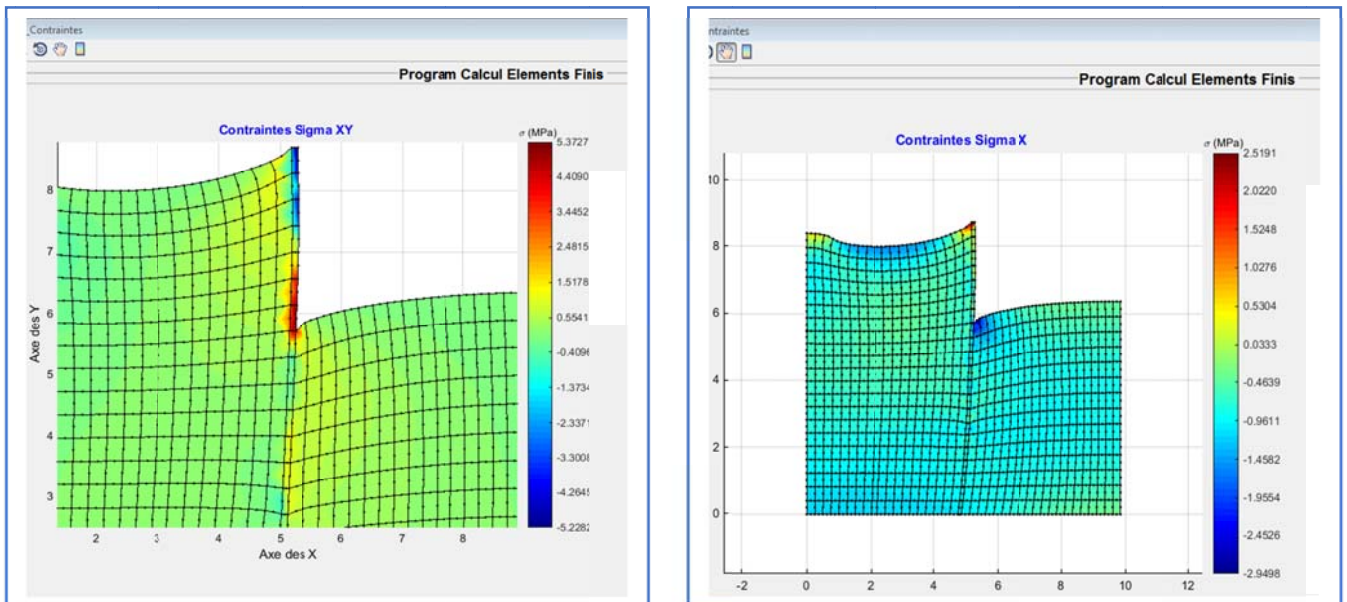


Figure 178 distribution des contraintes σ_{xy} σ_x avec Palplanche

Cas sans Palplanche

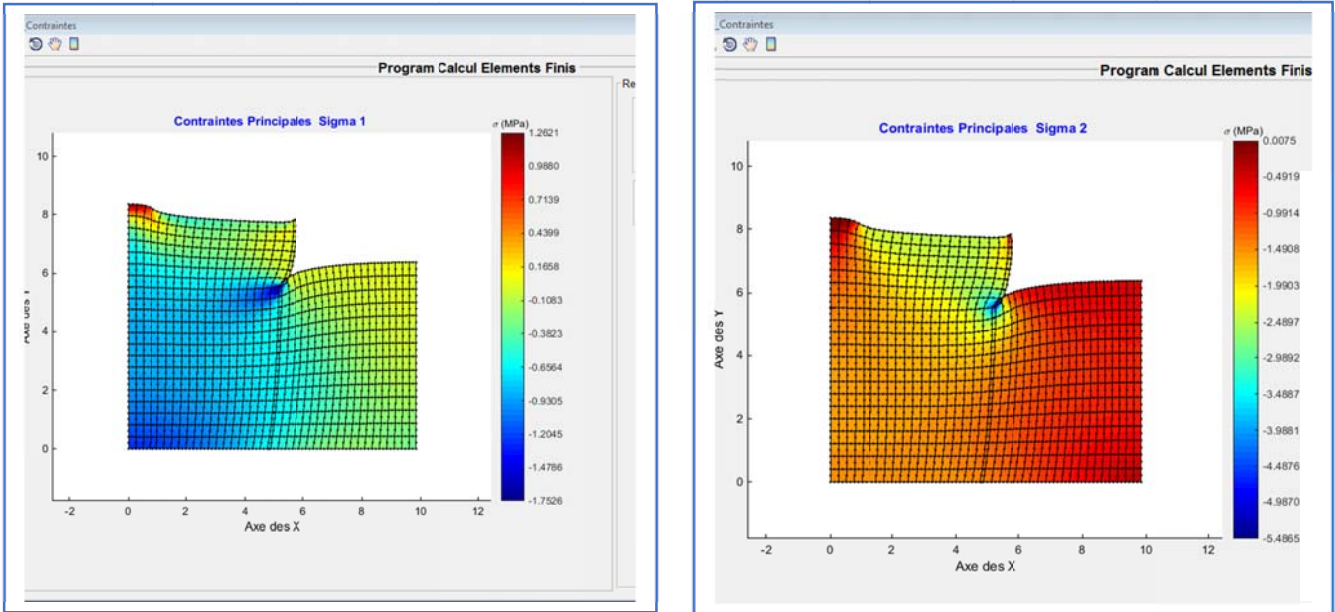


Figure 179 : distribution des contraintes Principale $\sigma_1 \sigma_2$

Cas avec Palplanche

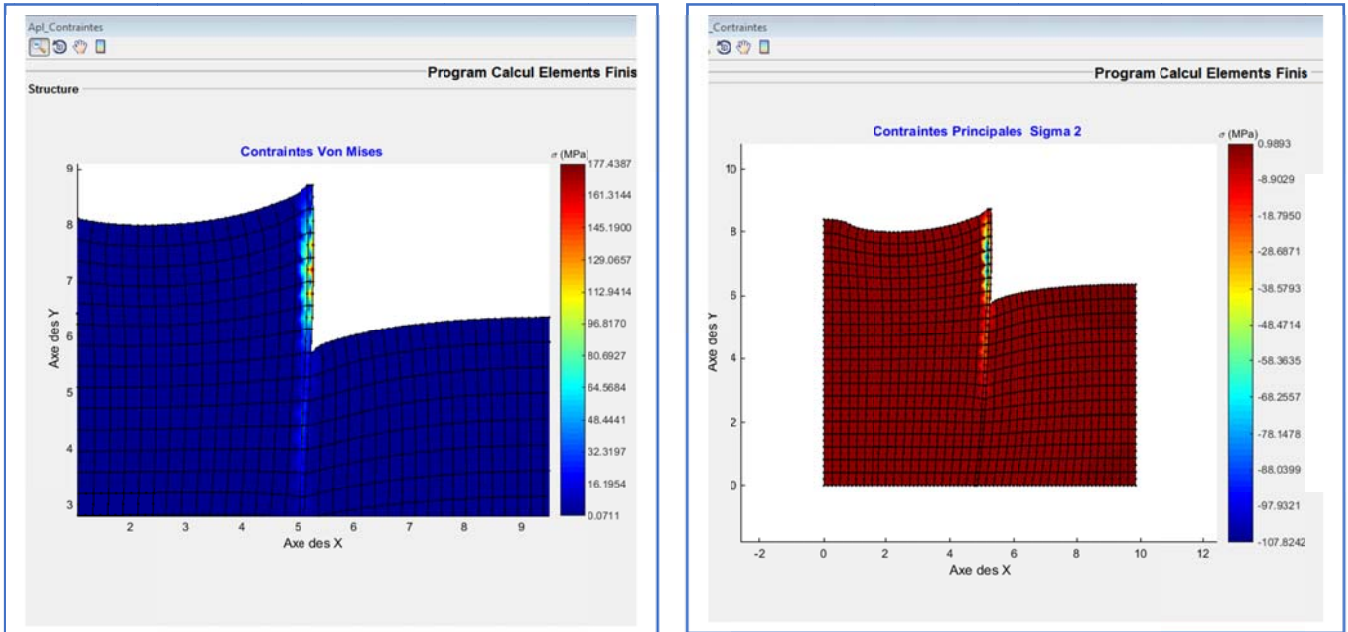


Figure 180 : distribution des contraintes Principale σ_2 et de von mises

5.3 Apl design T6

5.3.1 Exemple 01

Soit à étudier une plaque en acier en contrainte plane avec 436 éléments finis triangulaire T6 et 1047 nœuds ,encastrée sur les deux extrémités gauche et basse ,et soumise a une charge uniformément répartie sur extrémité droite.

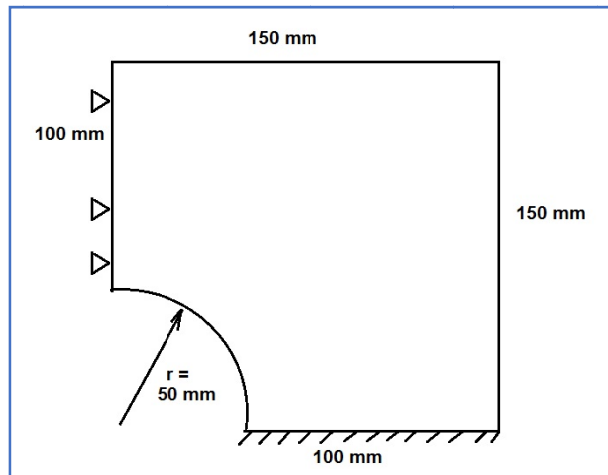


Figure 181

Schéma du maillage

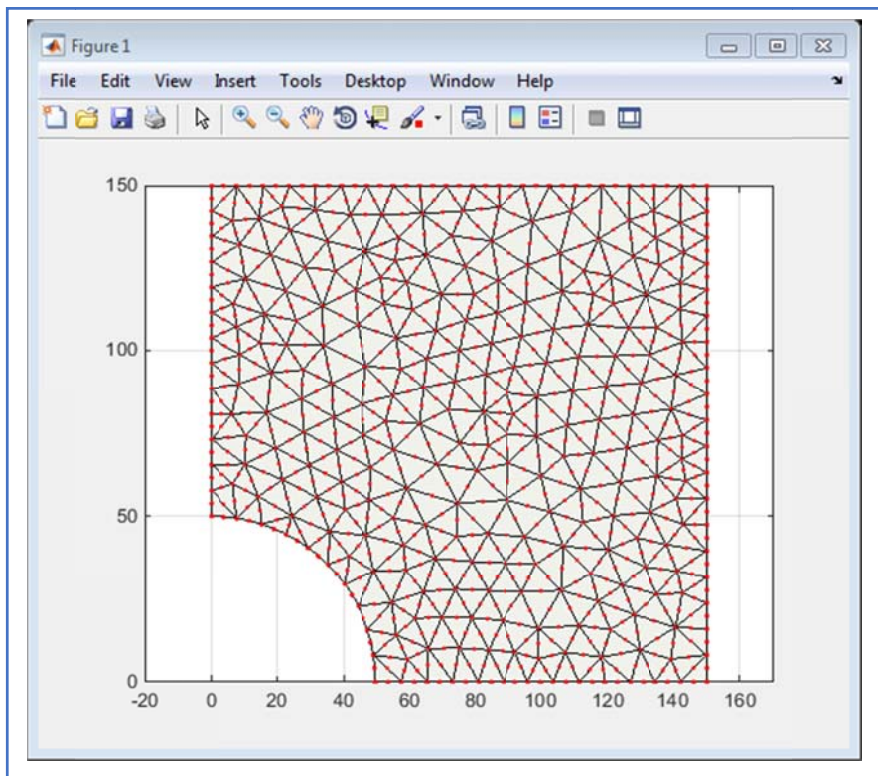


Figure 182 : maillage de la structure

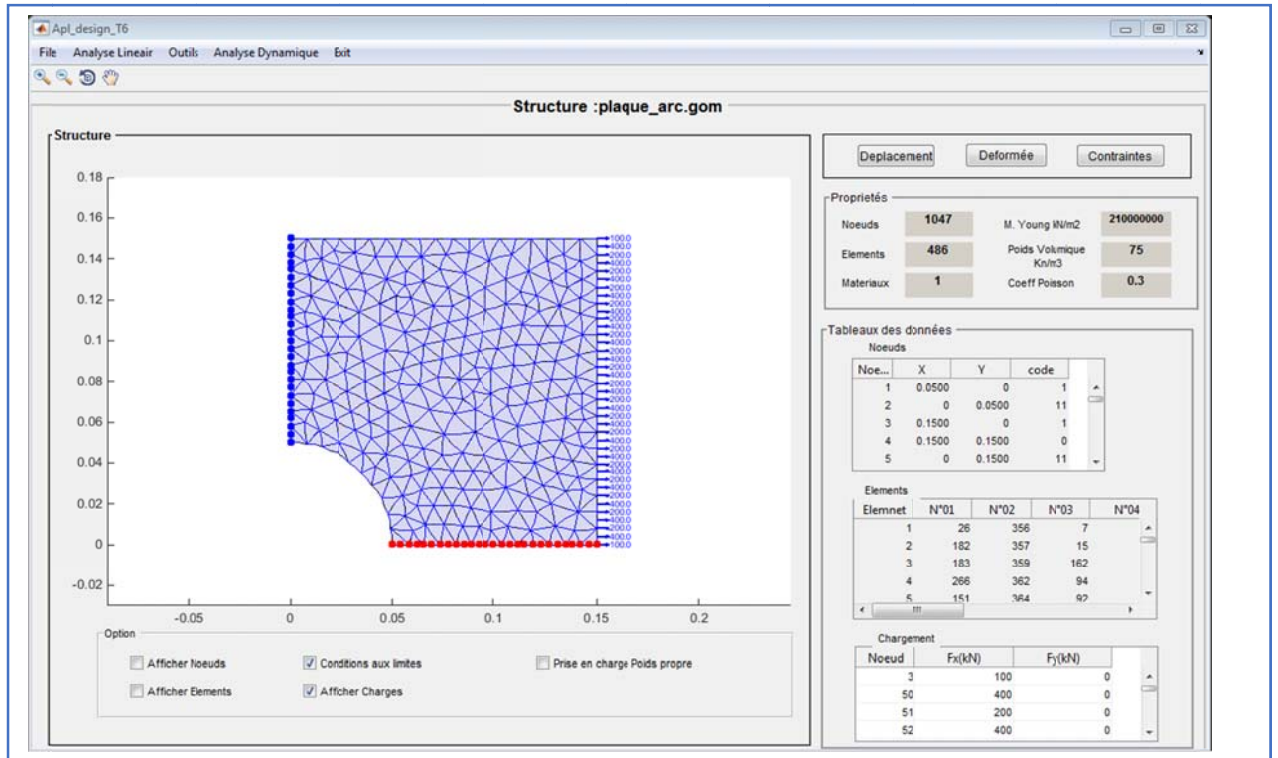


Figure 183 : Maillage et conditions aux limites

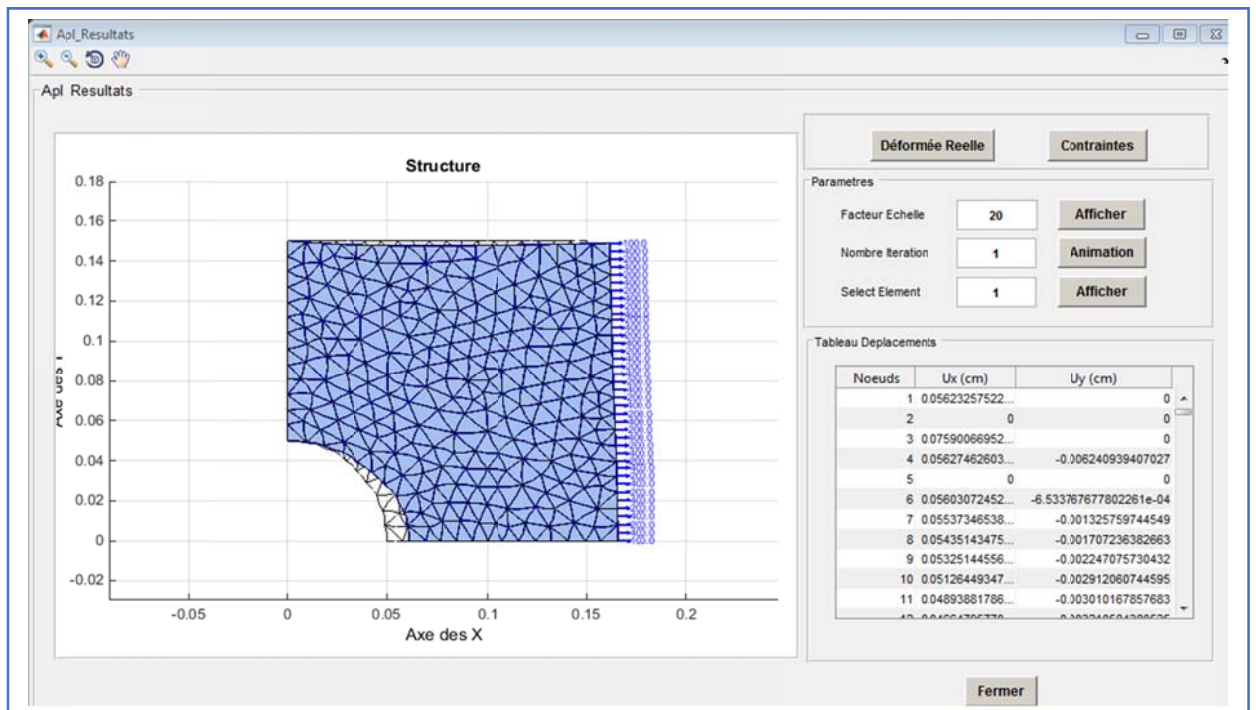


Figure 184 : déformée de la structure

Deplacement dans la structure (m)

Deplacement_Noeuds (m) **deplacement Max (m)**

	coord_X	coord_Y	deplc_X	deplc_Y
1	0.05	0.00	5.623257512000000e-04	0
2	0.00	0.05	0	0
3	0.15	0.00	7.590066913000000e-04	0
4	0.15	0.15	5.627462614000000e-04	-6.240939407000000e-05
5	0.00	0.15	0	0
6	0.05	0.00	5.603072412000000e-04	-6.533767678000000e-06
7	0.05	0.01	5.537346518000000e-04	-1.325759745000000e-05
8	0.05	0.01	5.435143415000000e-04	-1.707236383000000e-05
9	0.05	0.02	5.325144516000000e-04	-2.247075730000000e-05
10	0.05	0.02	5.126449317000000e-04	-2.912060745000000e-05
11	0.05	0.02	4.893881717000000e-04	-3.010167858000000e-05
12	0.04	0.03	4.664795710000000e-04	-3.210584381000000e-05
13	0.04	0.03	4.410849619000000e-04	-3.677341454000000e-05
14	0.04	0.03	4.133293628000000e-04	-3.754441569000000e-05
15	0.04	0.04	3.808797316000000e-04	-3.880531128000000e-05
16	0.03	0.04	3.441568619000000e-04	-3.821576300000000e-05
17	0.03	0.04	3.149356145000000e-04	-3.832916095000000e-05
18	0.03	0.04	2.687660962000000e-04	-3.657593769000000e-05
19	0.02	0.05	2.256014016000000e-04	-2.738483758000000e-05
20	0.02	0.05	1.890166470000000e-04	-1.981348888000000e-05
21	0.02	0.05	1.475500017000000e-04	-1.526352346000000e-05
22	0.01	0.05	1.097634470000000e-04	-8.420036472000000e-06
23	0.01	0.05	7.150741160000000e-05	2.495795653000000e-06
24	0.00	0.05	3.527226319000000e-05	4.726938589000000e-06

Figure 185 : résultats déplacements

On présente les figures les diagrammes des déplacements u_x et u_y

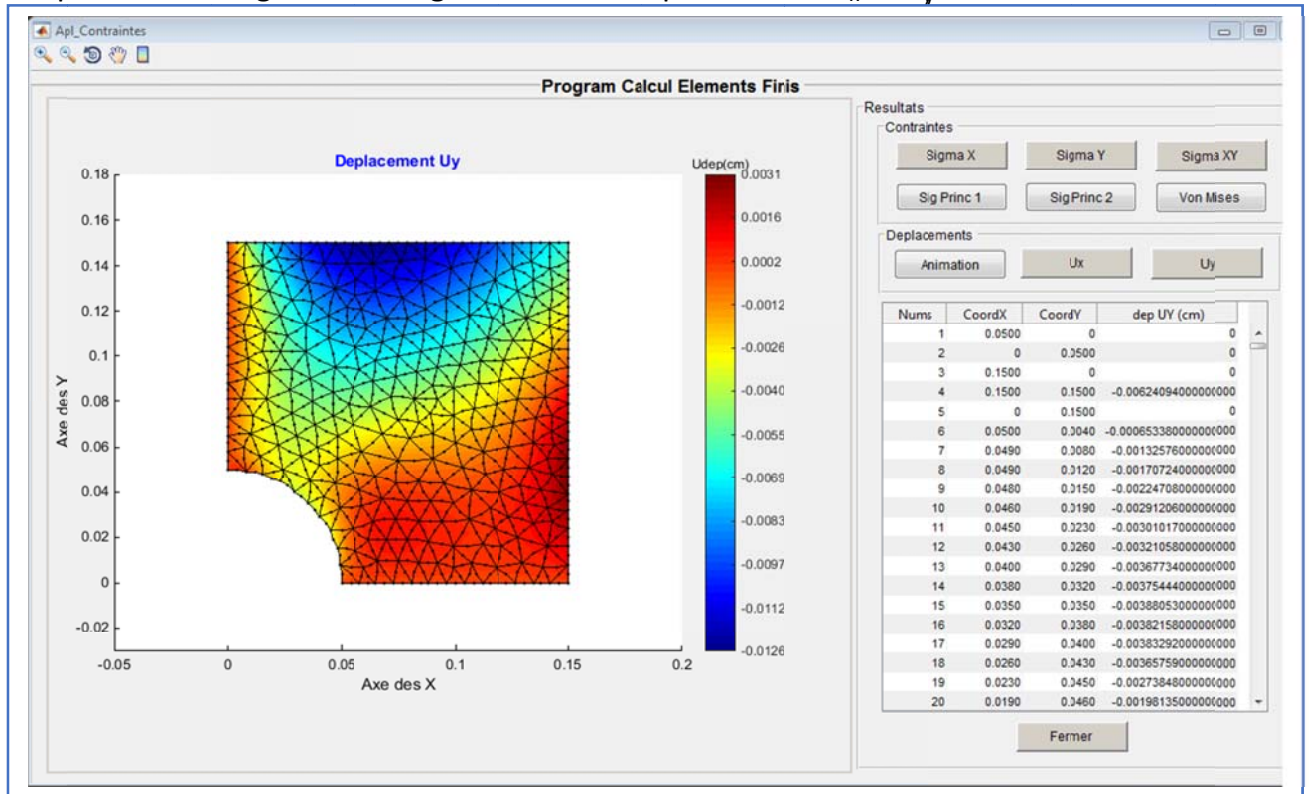


Figure 186 : déplacement suivant Uy

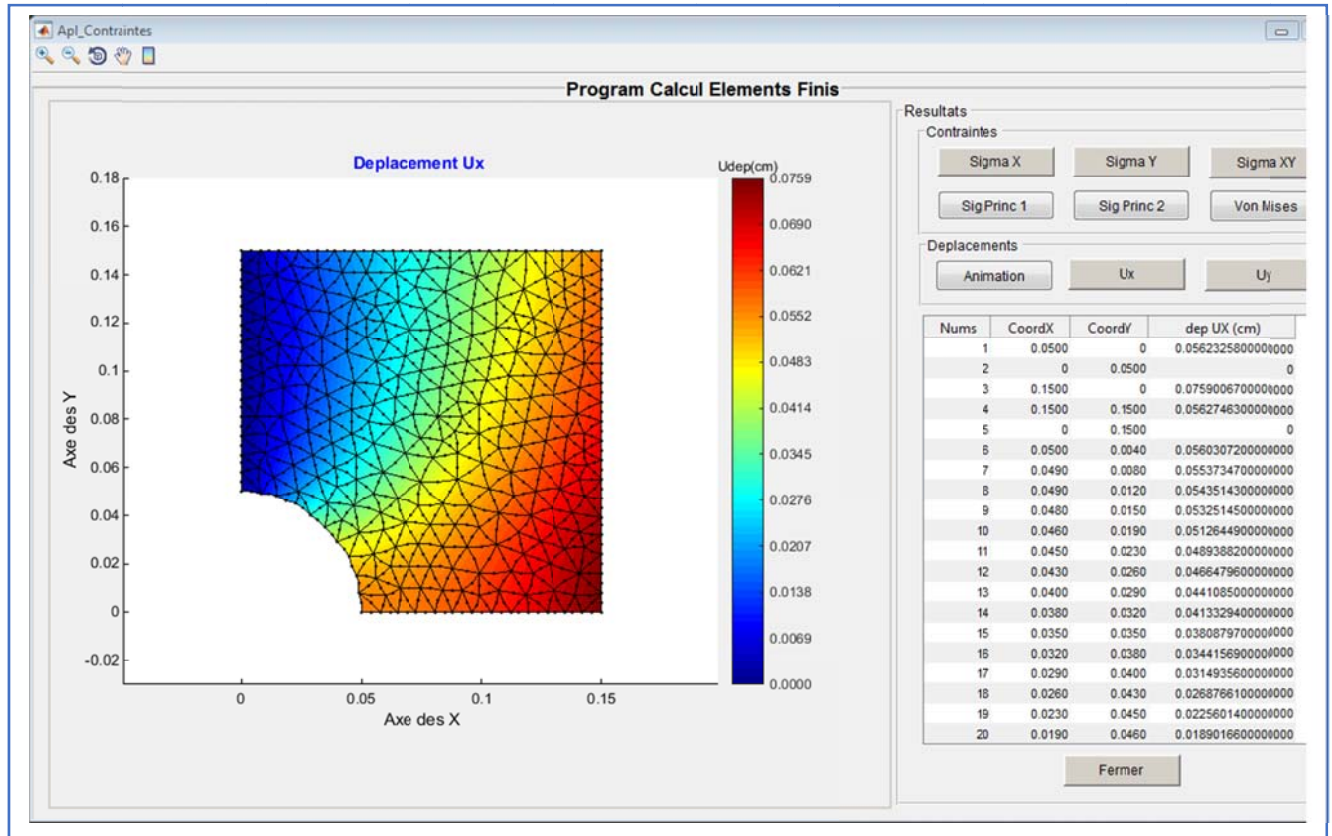


Figure 187 : déplacement suivant Ux

diagrammes des contraintes σ_x , σ_y et σ_{xy} et contraintes principales σ_2 et contraintes de von mises .

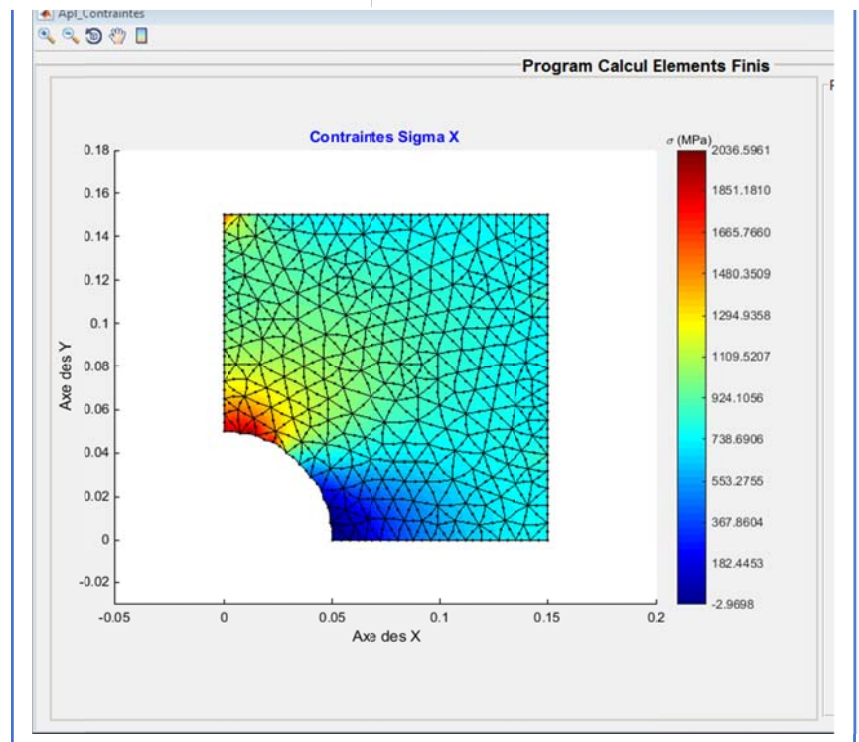


Figure 188 distribution des contraintes σ_x

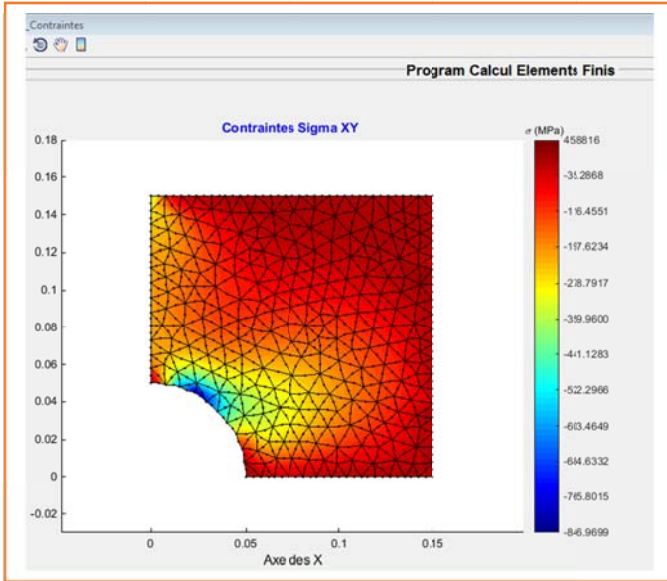


Figure 189 distribution des contraintes σ_{XY}

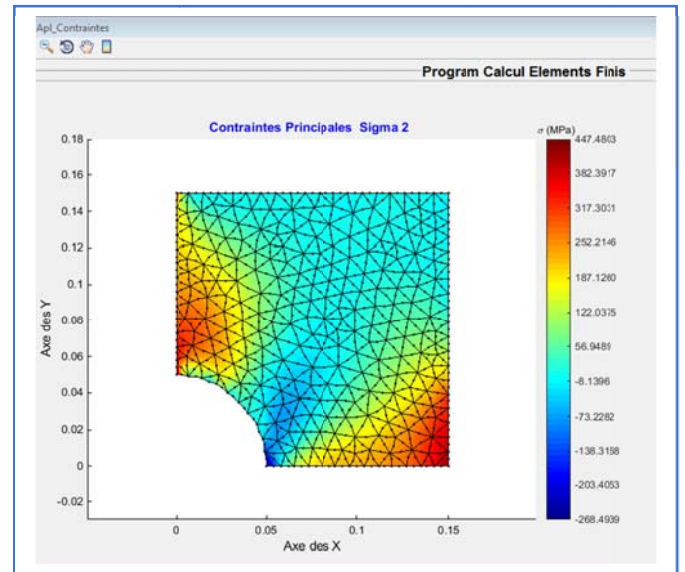


Figure 190 distribution des contraintes σ_z

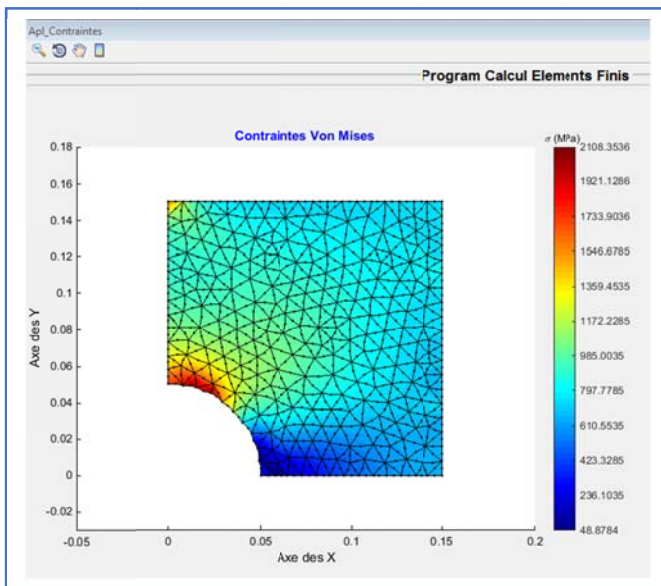


Figure 191 distribution des contraintes $\sigma_{von\ mises}$

ontraintes dans la structure KN/m2

	Contraintes aux Noeuds (MPa)			Contraintes centre Element (MPa)		
	X_Strs	Y_Strs	XY_Strs	X_Strs	Y_Strs	XY_Strs
1	0.159414587900000	-2.68493768500000...	-0.166345656900000			
2	1.910218935000000e+03	4.489194321000000...	45.881551580000000			
3	7.493860390000000e+02	4.150320097000000...	-0.619396126400000			
4	7.499661860000000e+02	-0.033381356360000	0.033381356360000			
5	1.631128742000000e+03	4.010554963000000...	-3.739279764000000e+02			
6	-1.949961217000000	-2.408726349000000...	0.739121820300000			
7	6.388595631000000	-1.405987198000000...	-17.543388670000002			
8	-2.969758044000000	-90.69699856000000010	-27.148120140000000			
9	53.793643500000000	39.818454940000000	-83.604010660000000			
10	98.808107340000000	1.464335629000000...	-1.266513740000000e+02			
11	2.267102521000000e+02	2.856378084000000...	-2.483007911000000e+02			
12	3.115205500000000e+02	3.825082993000000...	-3.346174684000000e+02			
13	4.145813062000000e+02	4.128241085000000...	-4.030710524000000e+02			
14	5.168248946000000e+02	4.827549036000000...	-4.815126992000000e+02			
15	6.990710830000000e+02	5.1305809370000001...	-5.700888212000000e+02			
16	8.210354355000000e+02	5.250430988000000...	-6.444891378000000e+02			
17	1.042189585000000e+03	5.458124668000000...	-7.209165806000000e+02			
18	1.495372739000000e+03	6.289617235000000...	-8.469698565000000e+02			
19	1.759353419000000e+03	4.654659723000000...	-8.048190009000000e+02			
20	1.634238009000000e+03	2.585142598000000...	-6.471592947000000e+02			
21	1.826798851000000e+03	2.717626069000000...	-5.981095587000000e+02			
22	2.036596114000000e+03	2.372747548000000...	-4.913805862000000e+02			
23	1.824647969000000e+03	1.109517333000000...	-2.816439604000000e+02			
24	1.842563772000000e+03	2.424110105000000...	-86.095207830000000			
25	4.563652588000000	-1.531962359000000	-3.680963534000000			

5.3.2 Exemple 02

Soit à étudier un talus en déformation plane avec 507 éléments finis triangulaire T6 et 1108 nœuds, encastree a sa base et bloqué dans le sens x a ses deux extrémités, soumise à une charge uniformément répartie comme le montre la figure.

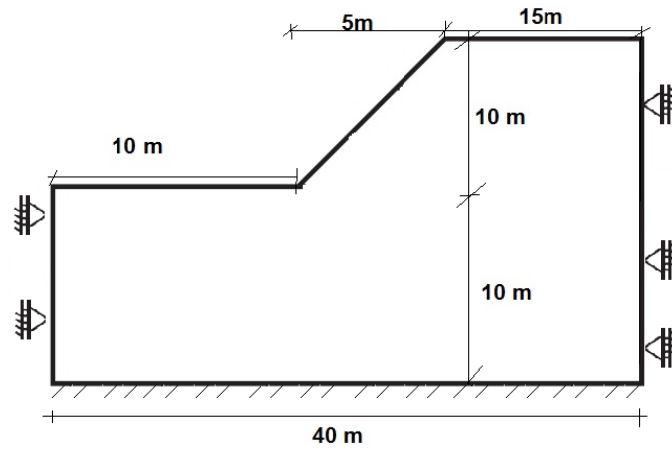


Figure 192 exemple 02

Schéma du maillage

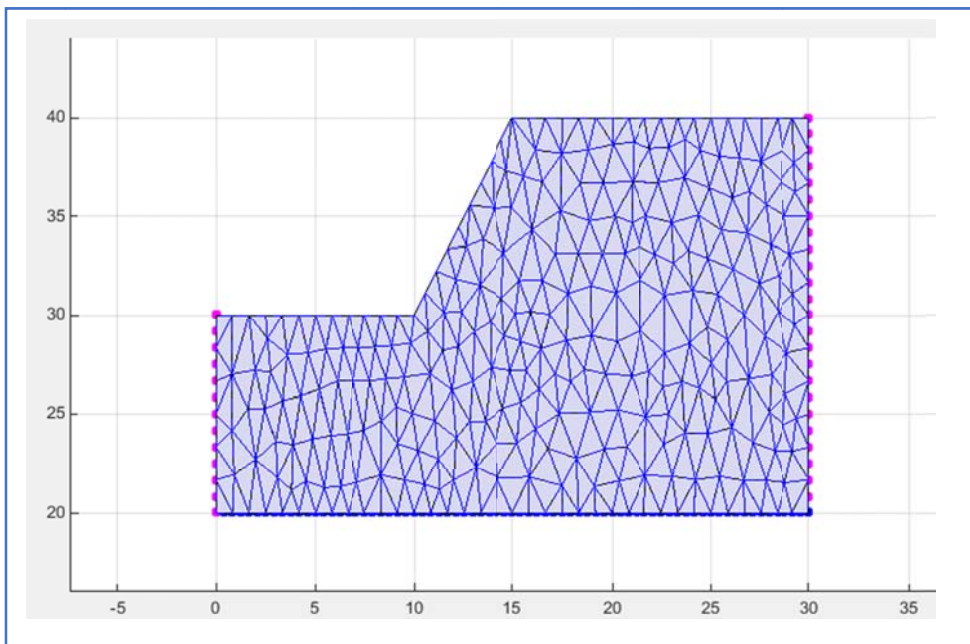


Figure 193 : maillage de la structure

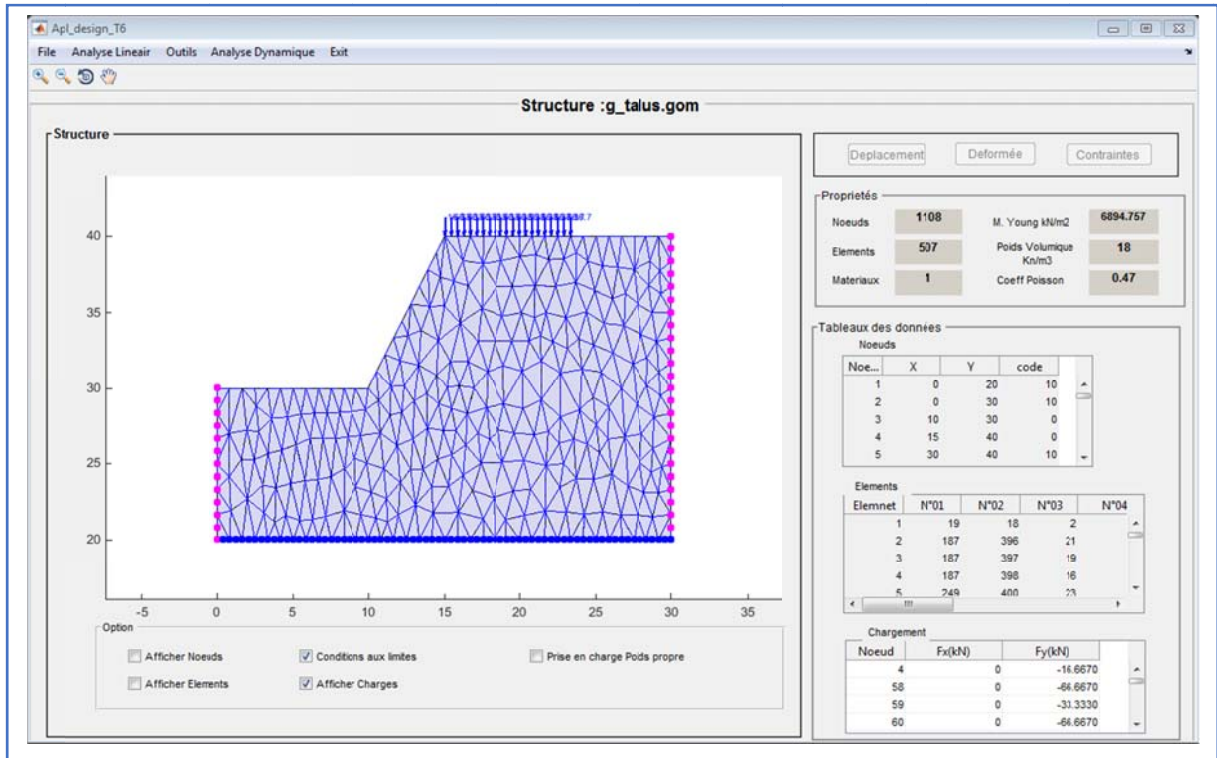


Figure 194 : Maillage et conditions aux limites

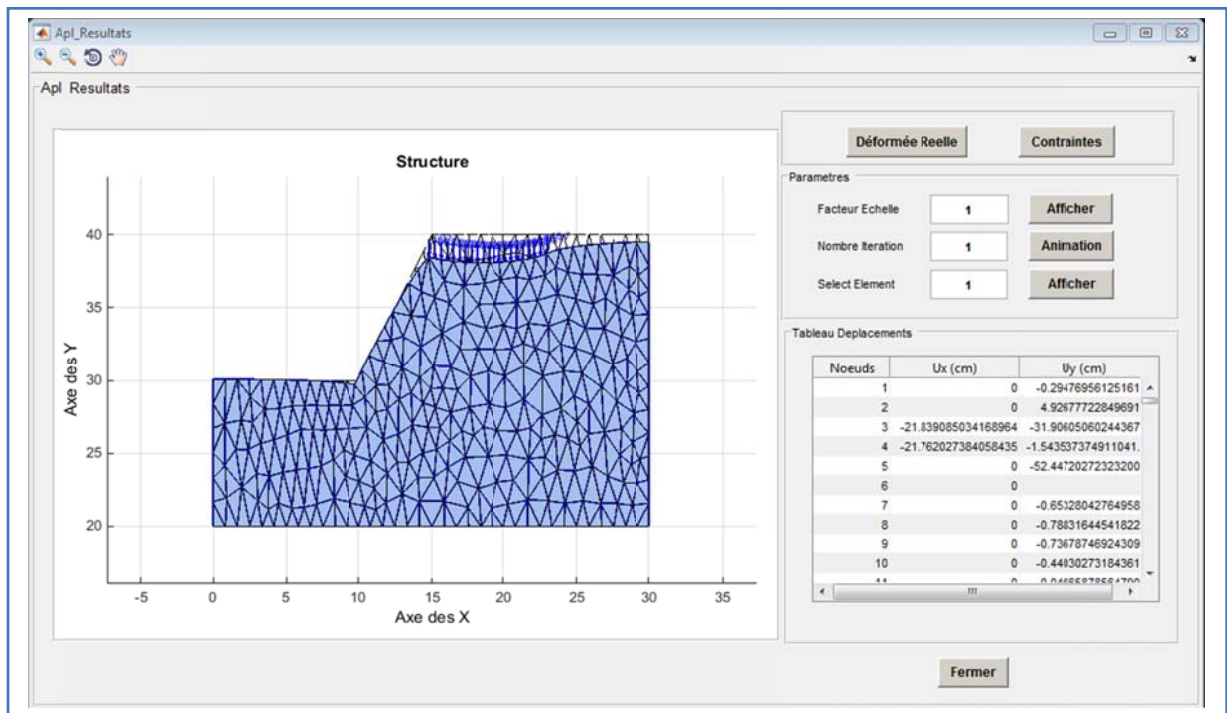


Figure 195 : déformée de la structure

Deplacement dans la structure (m)

Deplacement_Noeuds (m)		deplacement Max (m)		
	coord_X	coord_Y	deplc_X	deplc_Y
49	12.50	35.00	-0.425751972400000	-0.817166219100000
50	12.78	35.56	-0.432165307500000	-0.879288376200000
51	13.06	36.11	-0.435414904800000	-0.942488445100000
52	13.34	36.67	-0.435068224400000	-1.009744343000000
53	13.61	37.22	-0.430795722300000	-1.078563989000000
54	13.89	37.78	-0.422724280900000	-1.151387148000000
55	14.17	38.33	-0.401117260700000	-1.224621935000000
56	14.45	38.89	-0.360816610400000	-1.308293907000000
57	14.72	39.44	-0.300649915900000	-1.412116124000000
58	15.42	40.00	-0.199810649000000	-1.650370024000000
59	15.84	40.00	-0.193078281500000	-1.729572621000000
60	16.25	40.00	-0.194541279200000	-1.789288836000000
61	16.67	40.00	-0.198909059900000	-1.837489555000000
62	17.09	40.00	-0.205729496400000	-1.873101871000000
63	17.50	40.00	-0.214080868100000	-1.900316472000000
64	17.92	40.00	-0.222958122500000	-1.919546367000000
65	18.34	40.00	-0.232476388200000	-1.931963155000000
66	18.75	40.00	-0.242565002300000	-1.935582114000000
67	19.17	40.00	-0.253007434200000	-1.932344040000000
68	19.59	40.00	-0.264548659100000	-1.921600362000000
69	20.00	40.00	-0.276262514000000	-1.903981070000000
70	20.42	40.00	-0.288582261300000	-1.877602264000000
71	20.84	40.00	-0.301041490400000	-1.843466889000000
72	21.25	40.00	-0.315169953500000	-1.799045790000000
73	21.67	40.00	-0.329580208800000	-1.744114212000000
74	22.09	40.00	-0.342640019800000	-1.673551554000000
75	22.50	40.00	-0.354259462000000	-1.590953477000000
76	22.92	40.00	-0.360160700000000	-1.473404430000000
77	23.34	40.00	-0.361583269200000	-1.307842119000000

Figure 196 : résultats déplacements

On présente les figures les diagrammes des déplacements u_x et u_y

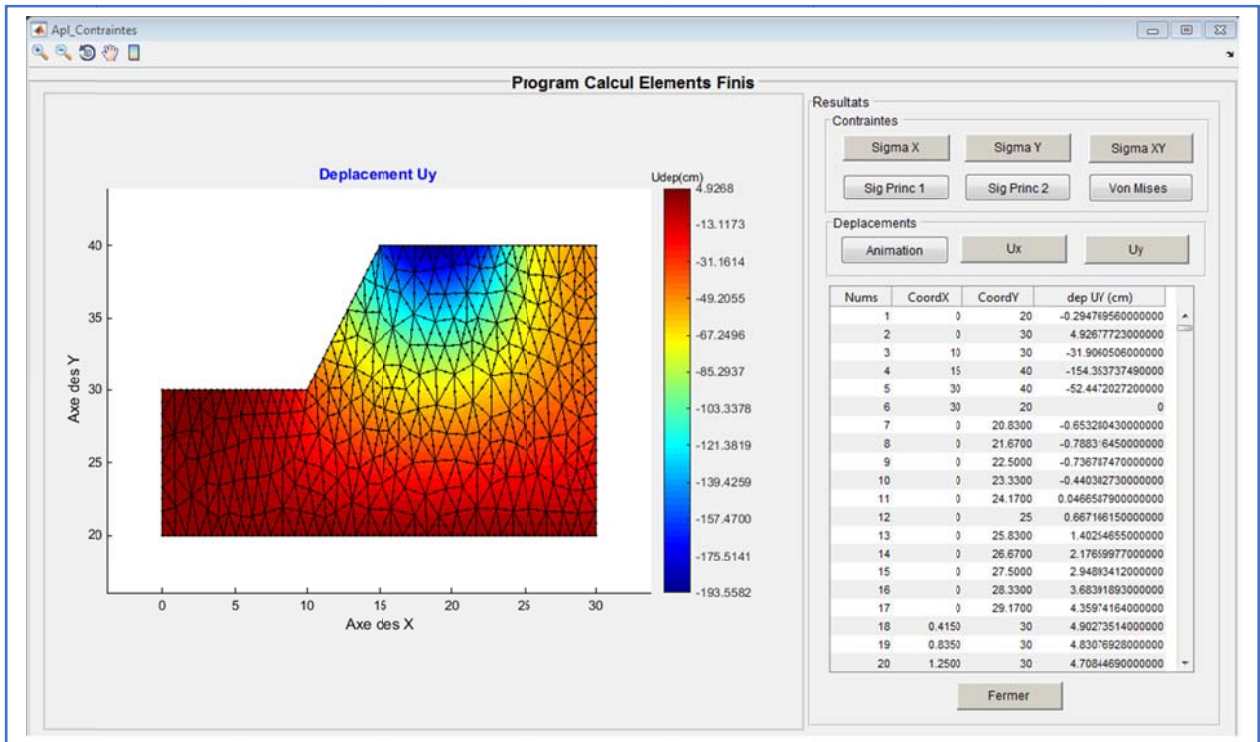


Figure 197 : déplacement suivant U_y

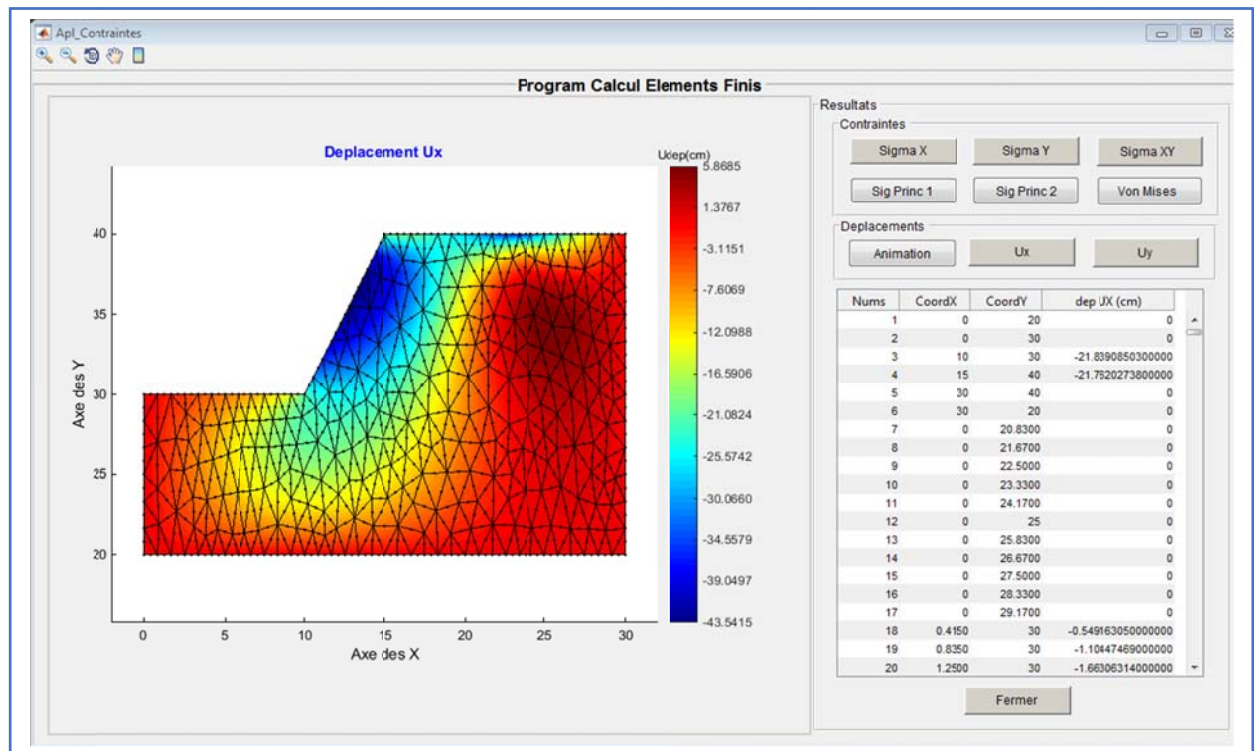


Figure 198 : déplacement suivant U_x

diagramme des contraintes σ_x , σ_y et σ_{xy} et contraintes principales σ_2 et contraintes de von mises .

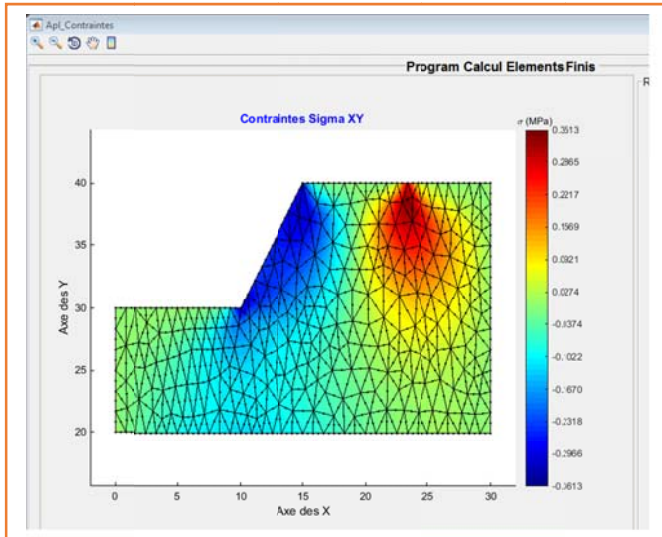


Figure 200 distribution des contraintes σ_{xy}

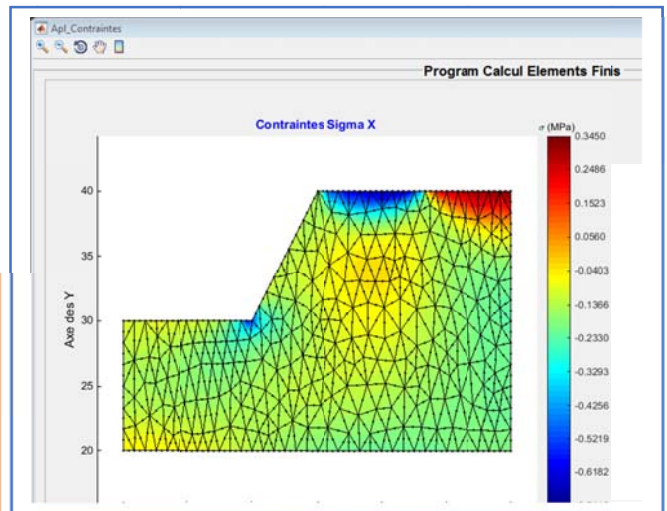


Figure 199 distribution des contraintes σ_x

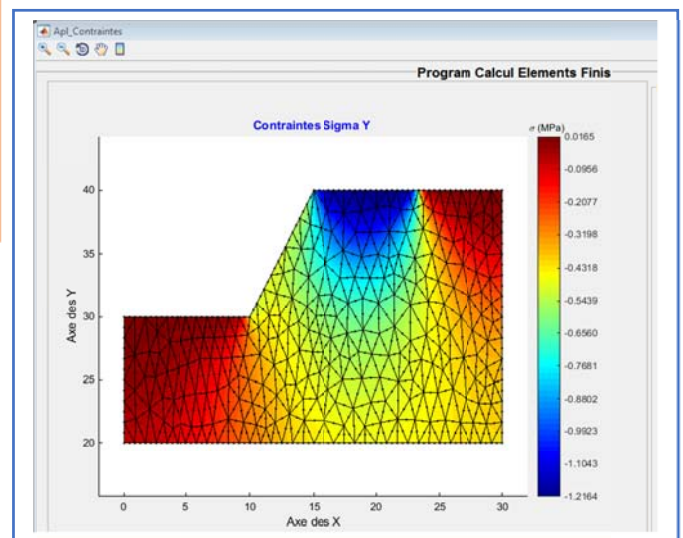


Figure 201 distribution des contraintes σ_y

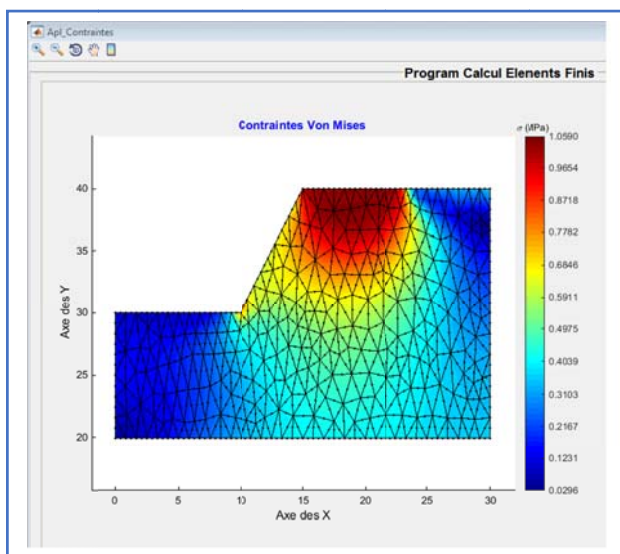


Figure 202 distribution des contraintes $\sigma_{von\ mises}$

Contraintes dans la structure KN/m2

	Contraintes aux Noeuds (MPa)		Contraintes centre Element (MPa)
	X_Strs	Y_Strs	XY_Strs
1	-0.34135414...	-0.058768547580000	0.012789743260000
2	-0.39703549...	1.188081864000000e-04	5.940409319000000e...
3	-0.33334228...	-0.433364945700000	-0.332482273800000
4	-0.18578863...	-0.896090141000000	-0.263508188600000
5	0.343101884...	2.906740951000000e-04	-1.453370475000000...
6	-0.19459554...	-0.409178860700000	3.970554882000000e...
7	-0.36210884...	-0.059098979060000	0.006282553198000
8	-0.39240323...	-0.053239431540000	-0.002086377950000
9	-0.11529734...	-0.041914544580000	-0.002078891677000
10	-0.13267951...	-0.031167496570000	-0.002355668973000
11	-0.14335528...	-0.021378323520000	-0.001278350489000
12	-0.15054707...	-0.013456924290000	-0.001340955921000
13	-0.14992296...	-0.007483966002000	-0.001001267925000
14	-0.14401613...	-0.003259910467000	-6.928653020000000...
15	-0.13636486...	-9.830234270000000e-...	-2.881786416000000...
16	-0.12327848...	-3.448557023000000e-...	-8.457448282000001...
17	-0.10574718...	2.251187450000000e-04	1.612439875000000e...
18	-0.39690816...	2.335638702000000e-04	2.144320904000000e...
19	-0.10231513...	3.880646930000000e-04	2.063040917000000e...
20	-0.10181266...	8.069816794000000e-04	4.844695188000000e...
21	-0.10391019...	2.766233230000000e-04	3.570090010000000e...
22	-0.39511621...	-5.137664112000000e-...	9.002149708000000e...
23	-0.39391979...	7.099267572999999e-05	0.001027440854000

5.4 Apl design Cub8

5.4.1 Exemple 01

Soit a calculer un dôme cylindrique d'une longueur de 20 m ayant un rayon de courbure de 6 m et d'une épaisseur de 0.5 m

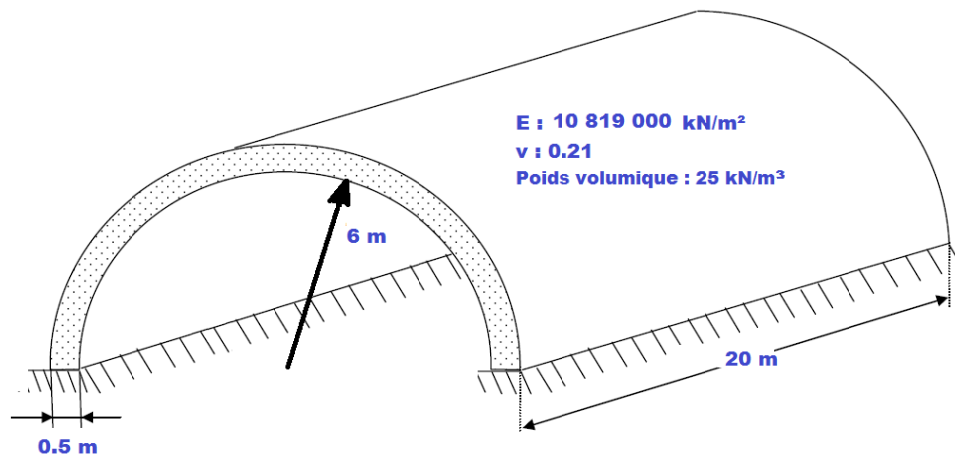


Figure 203 : structure a étudier

Sollicitations : Poids propre de la structure

Conditions aux limites : encastrement sur la base

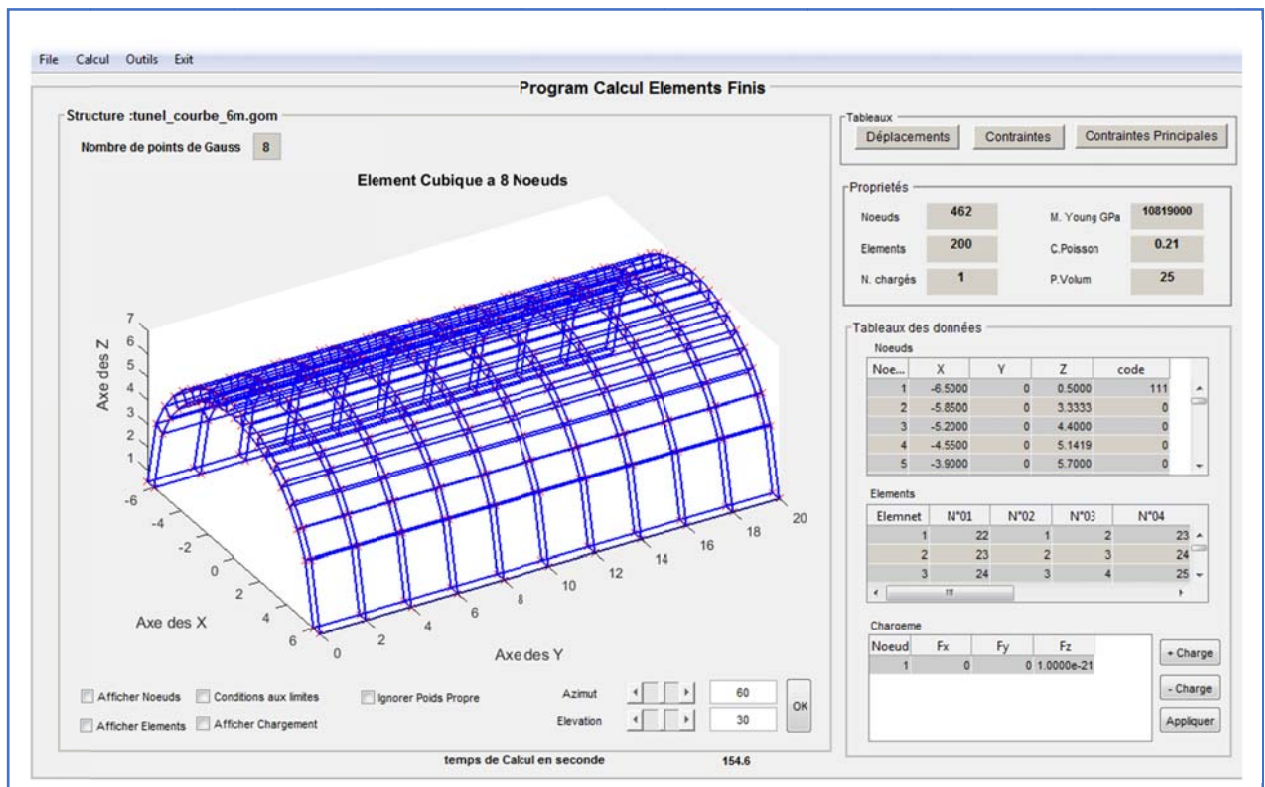


Figure 204 : maillage et affichage de la structure dans le programme

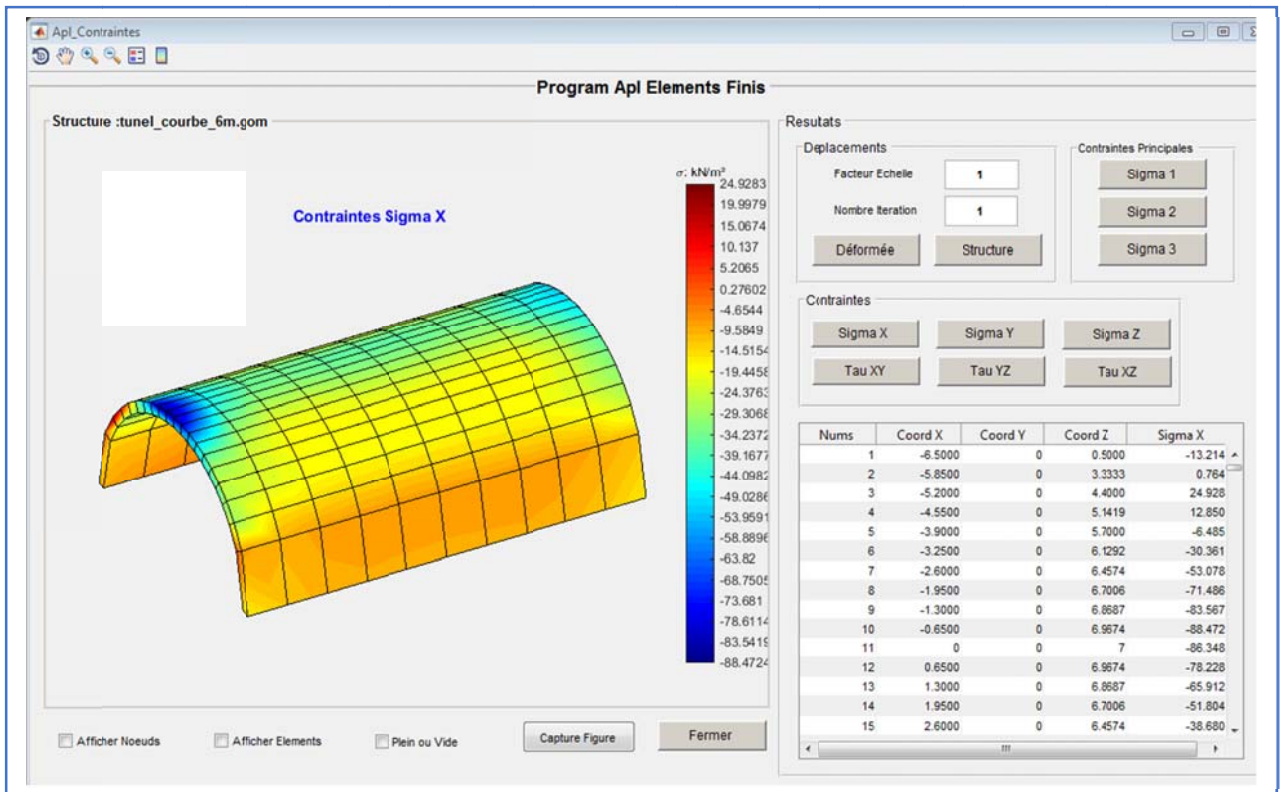


Figure 205 : distribution des contraintes σ_x

Résultats des contraintes

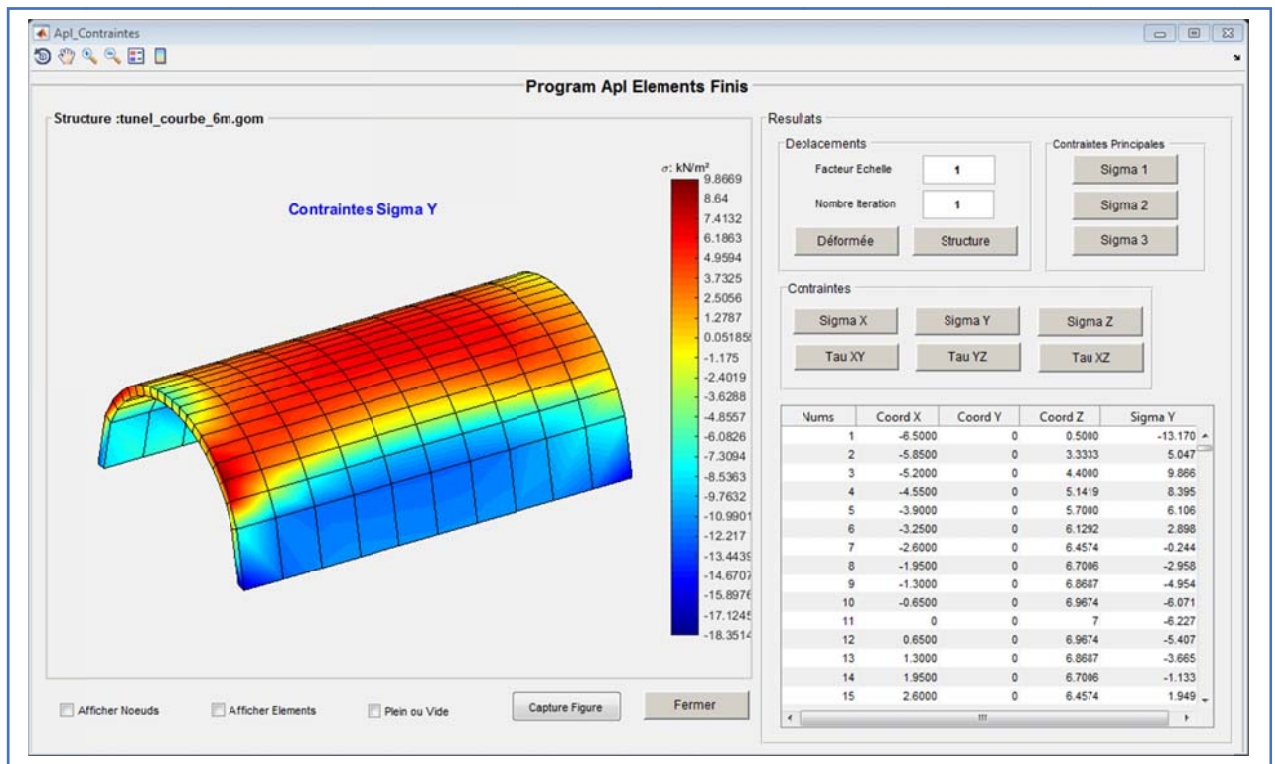


Figure 206 : distribution des contraintes σ_y

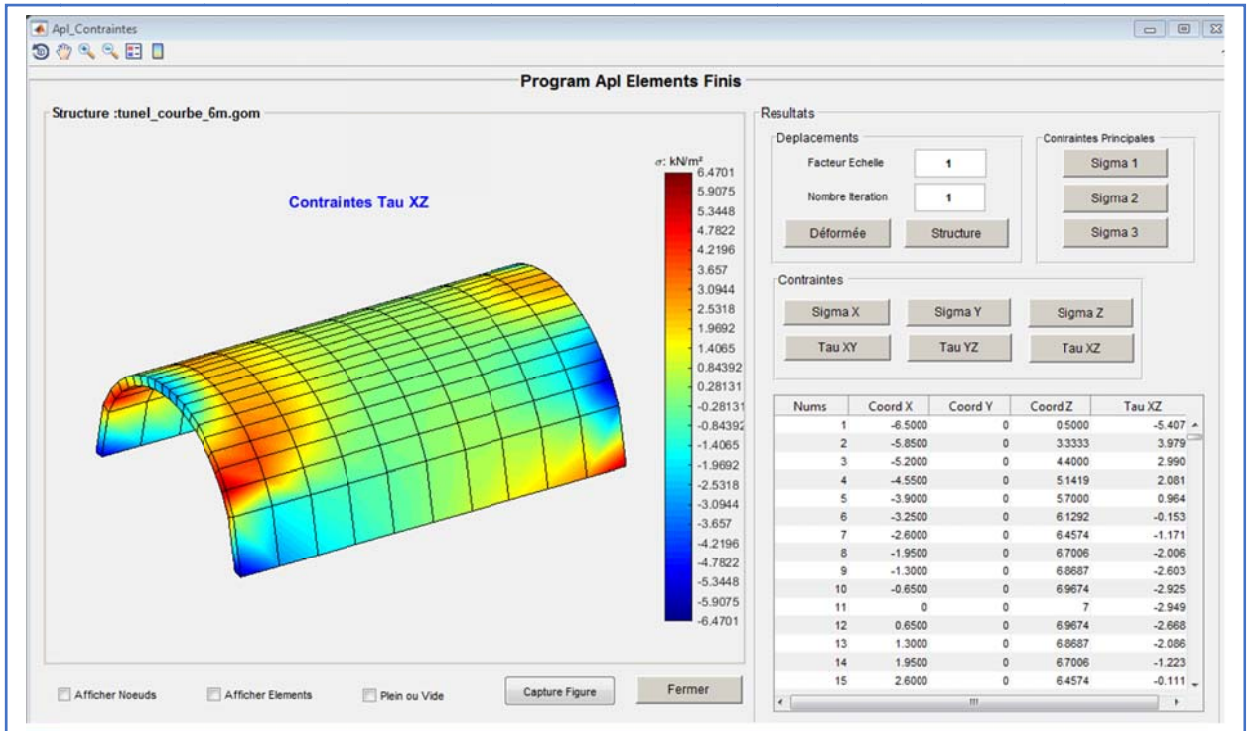


Figure 207 : distribution des contraintes σ_γ

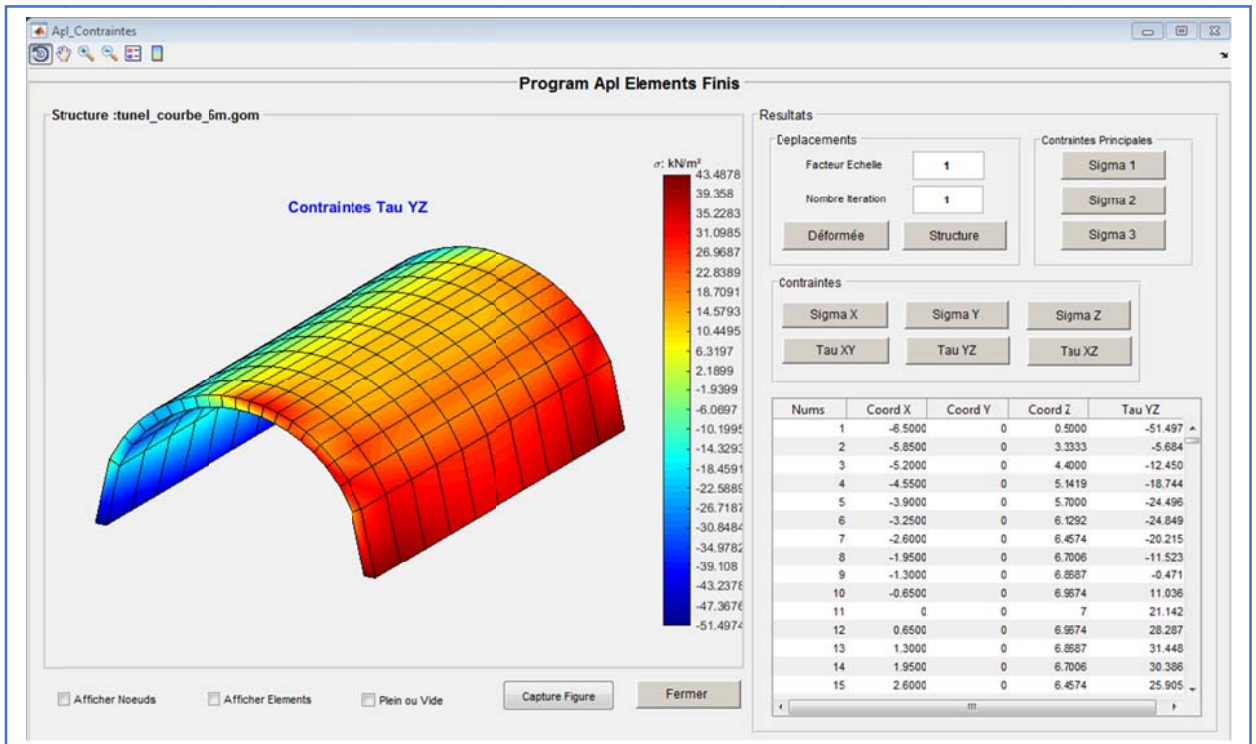


Figure 208 : distribution des contraintes σ_γ

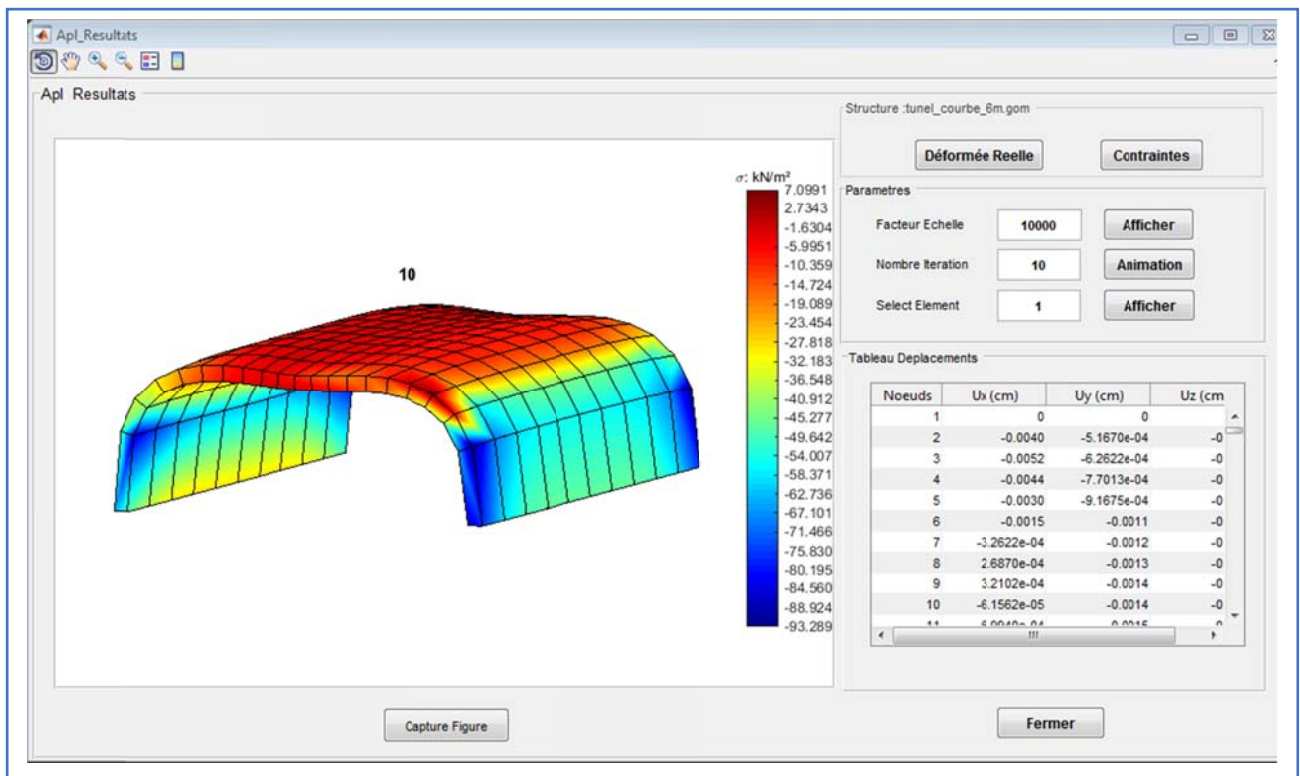
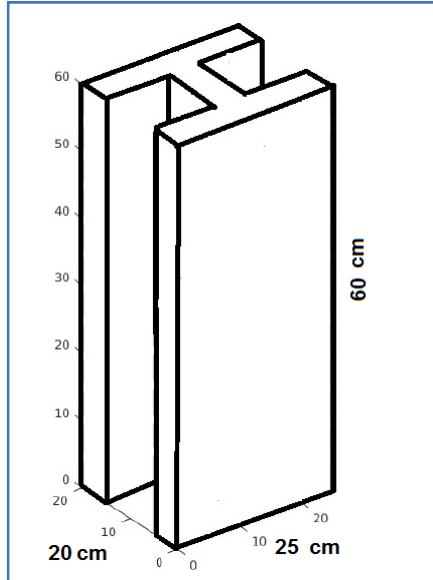


Figure 209 : déformée de la structure avec un facteur d'échelle

5.4.2 Exemple 02

Soit une poutre en H en acier de 60 cm de hauteur



$E = 210\,000\text{ MPa}$
 $\nu = 0.3$
 $P = 10\text{ kN}$

Figure 210 : structure Poutre en H

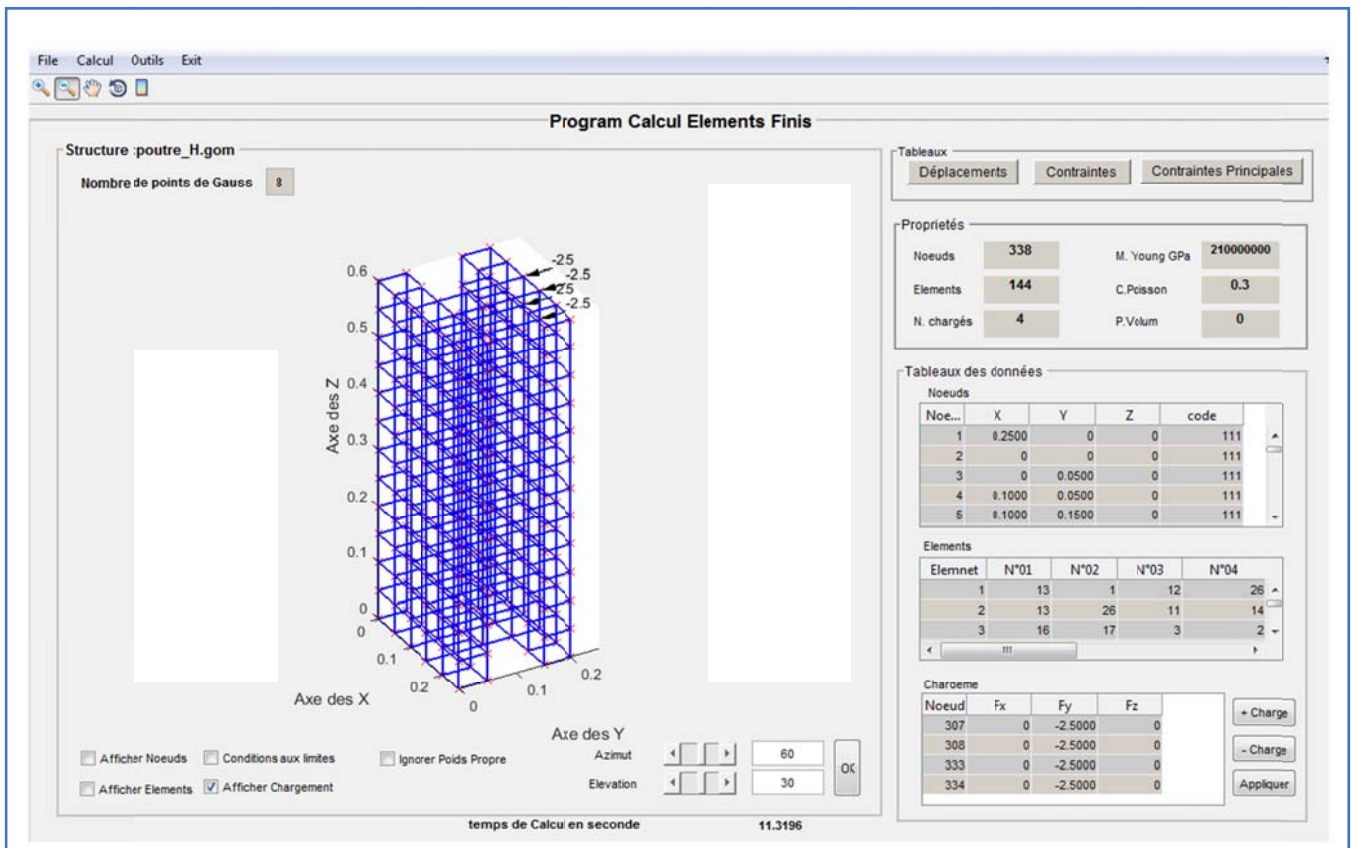


Figure 211 : maillage de la structure

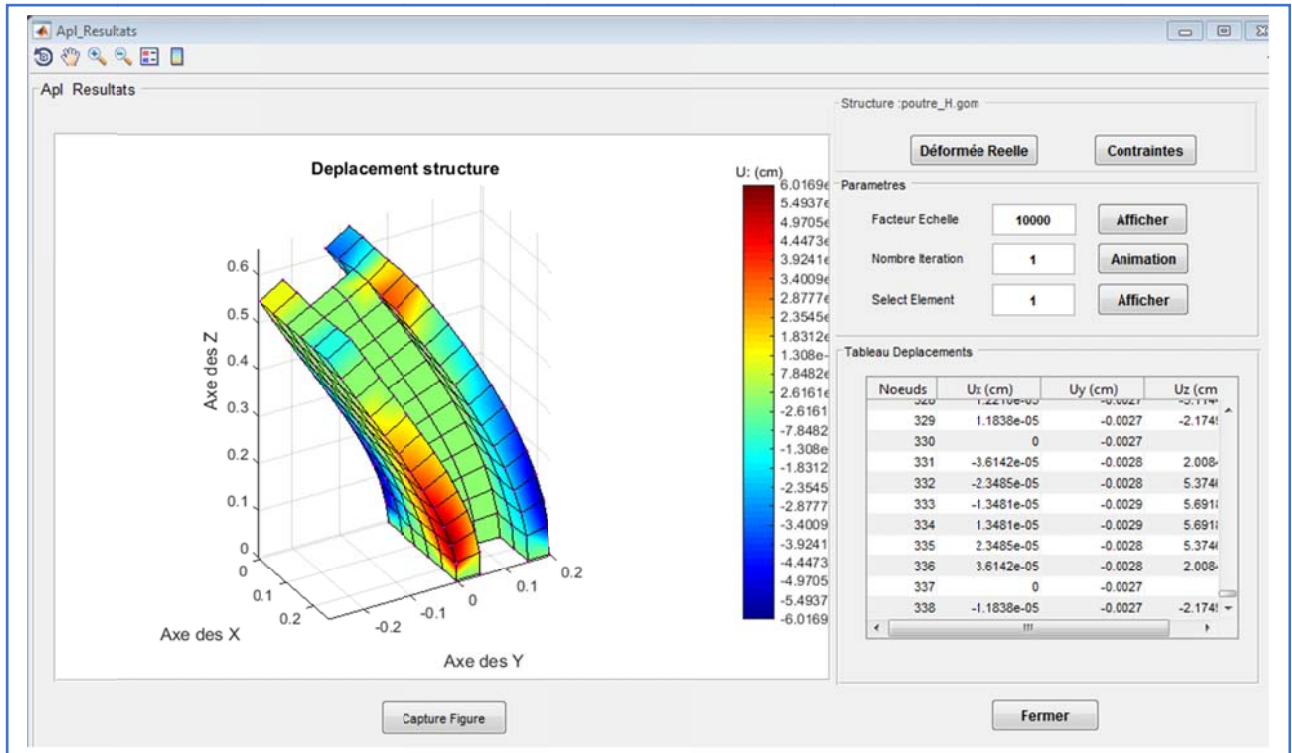


Figure 212 : déplacements de la structure U_x

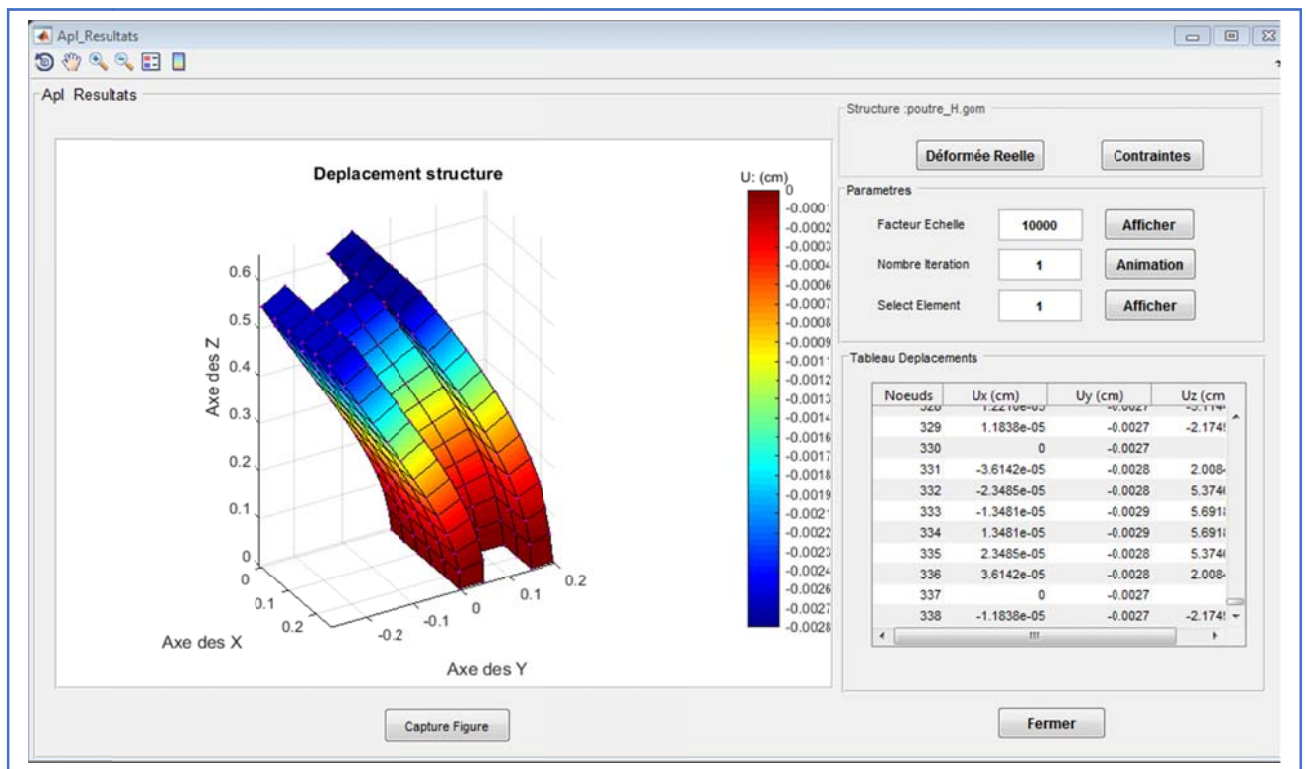


Figure 213 : déformée U_y de la structure

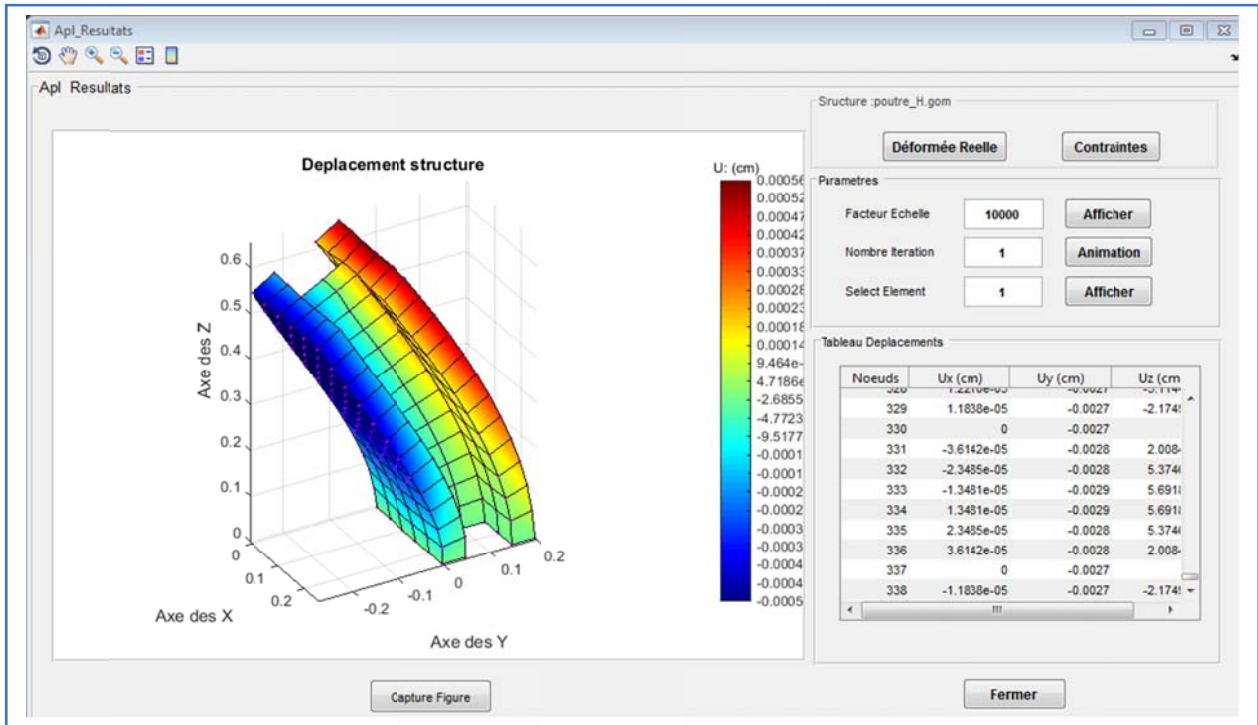


Figure 214 : déformée Uz de la structure

Diagramme des contraintes

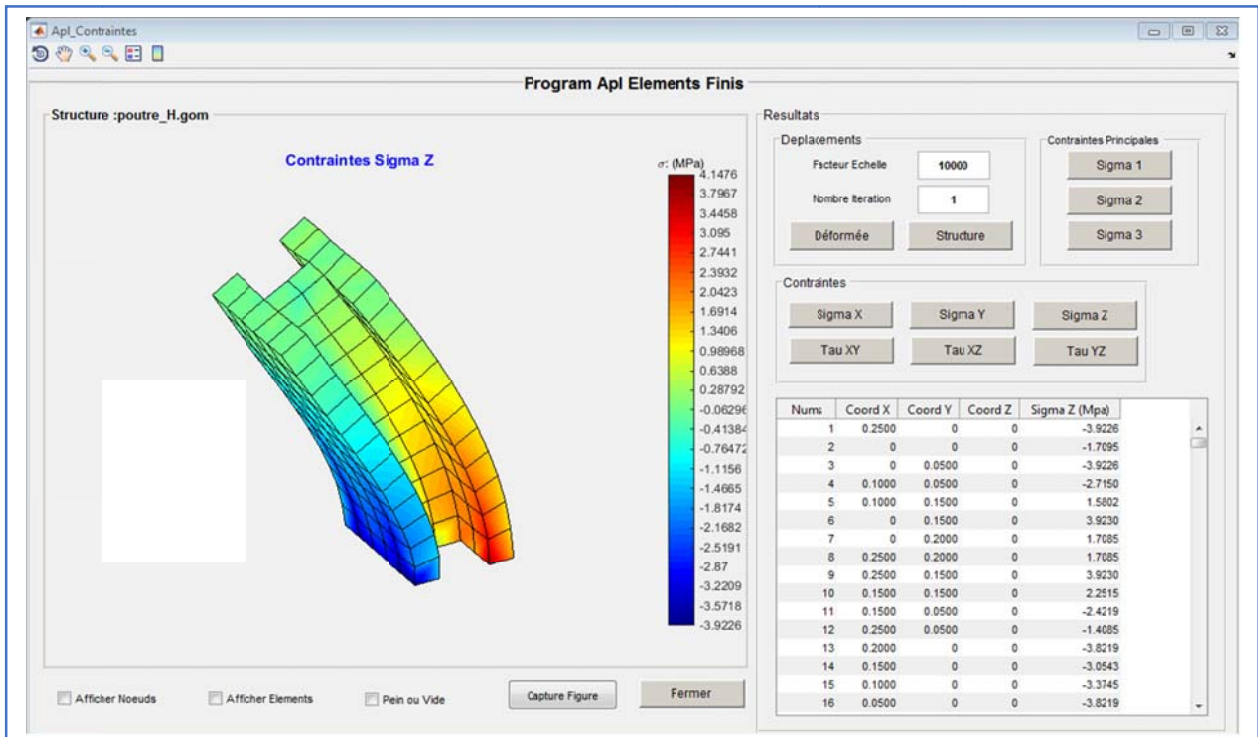


Figure 215 : distribution des contraintes σ_z

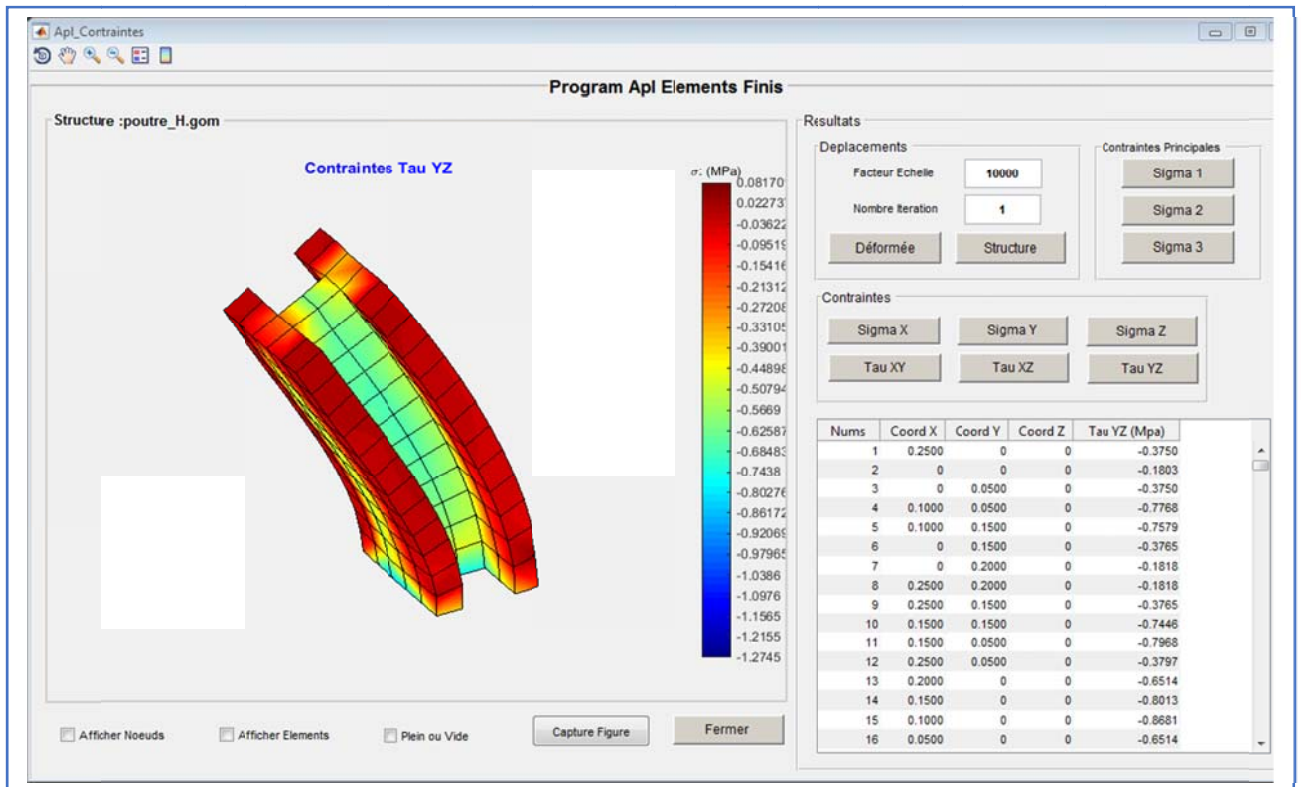


Figure 216 : distribution des contraintes σ_{yz}

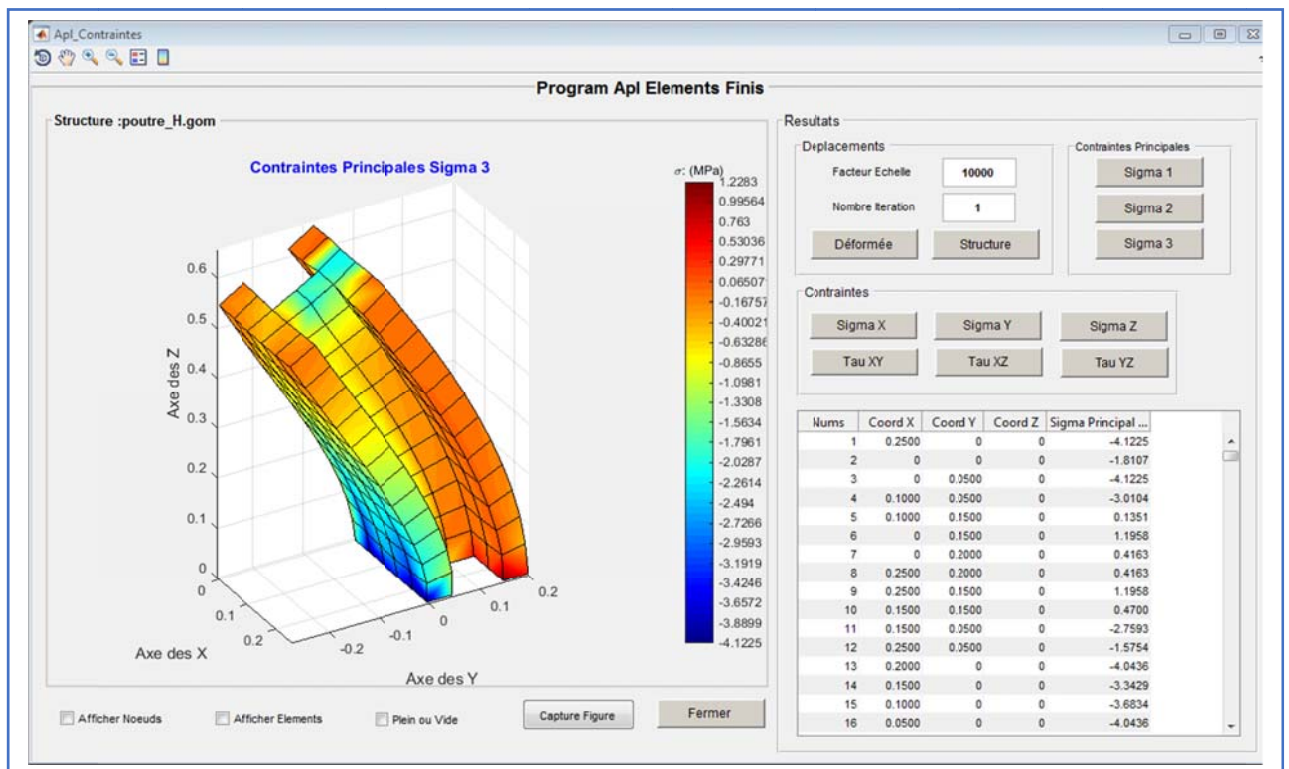


Figure 217 : distribution des contraintes $\sigma_{\text{principale } 3}$

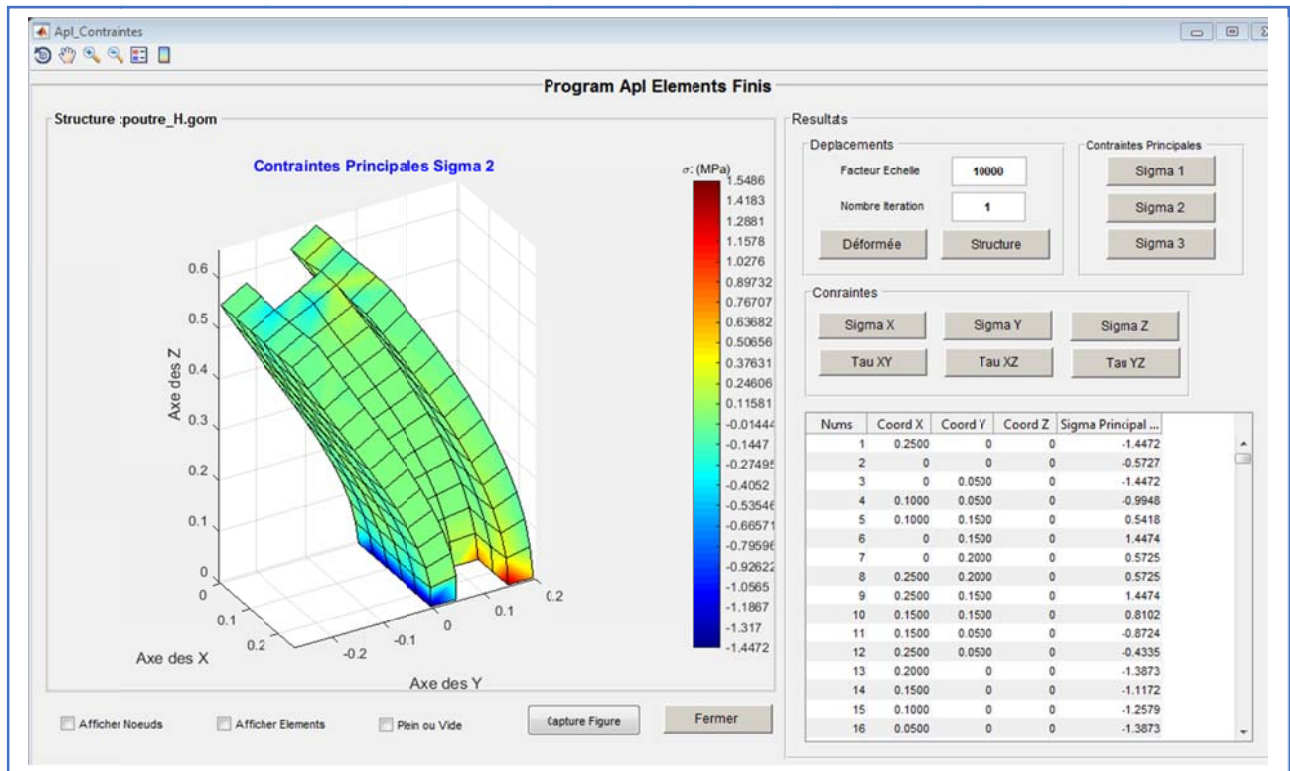


Figure 218 : distribution des contraintes σ principale 2

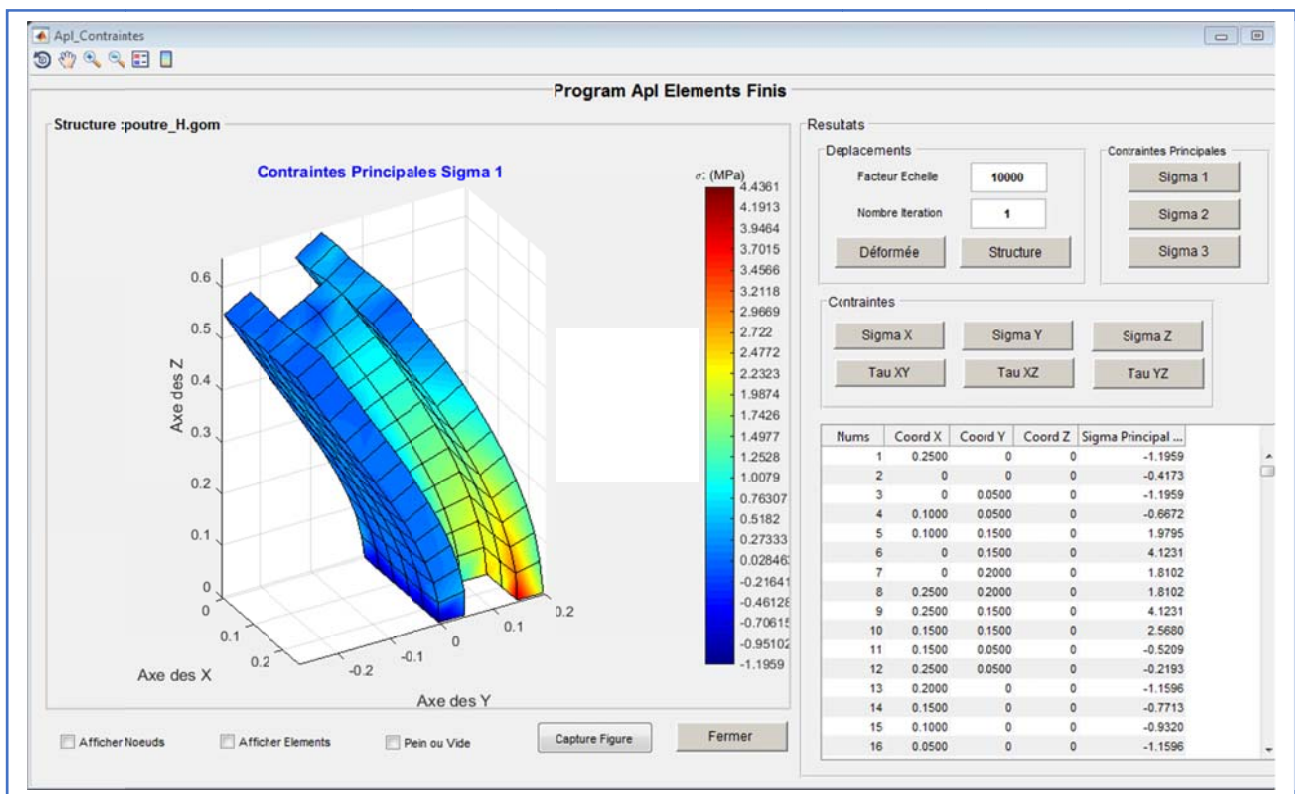


Figure 219 : distribution des contraintes σ principale 1

5.5 Apl design Hexa20

5.5.1 Exemple 01

Soit à calculer un dôme cylindrique d'une longueur de 20 m ayant un rayon de courbure de 6 m et d'une épaisseur de 0.5 m

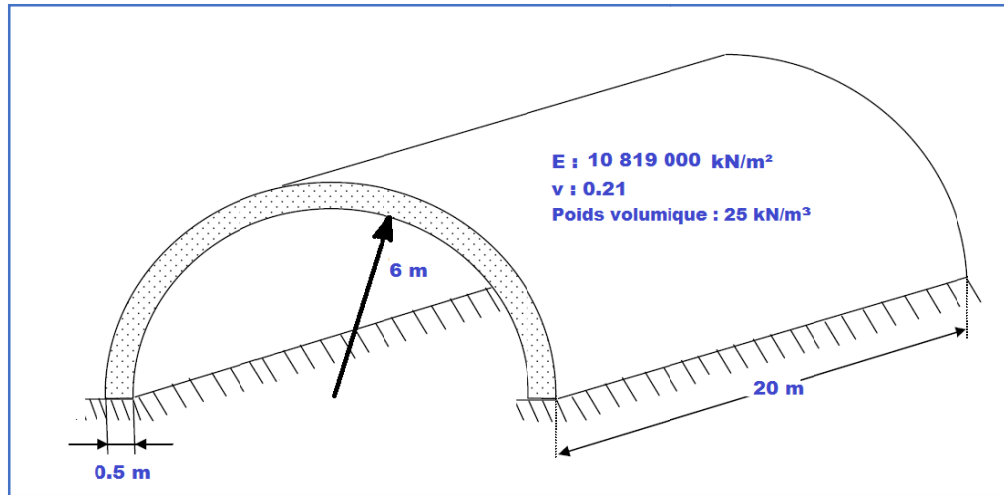


Figure 220 : structure à étudier

Sollicitations : Poids propre de la structure

Conditions aux limites : encastrement a la base

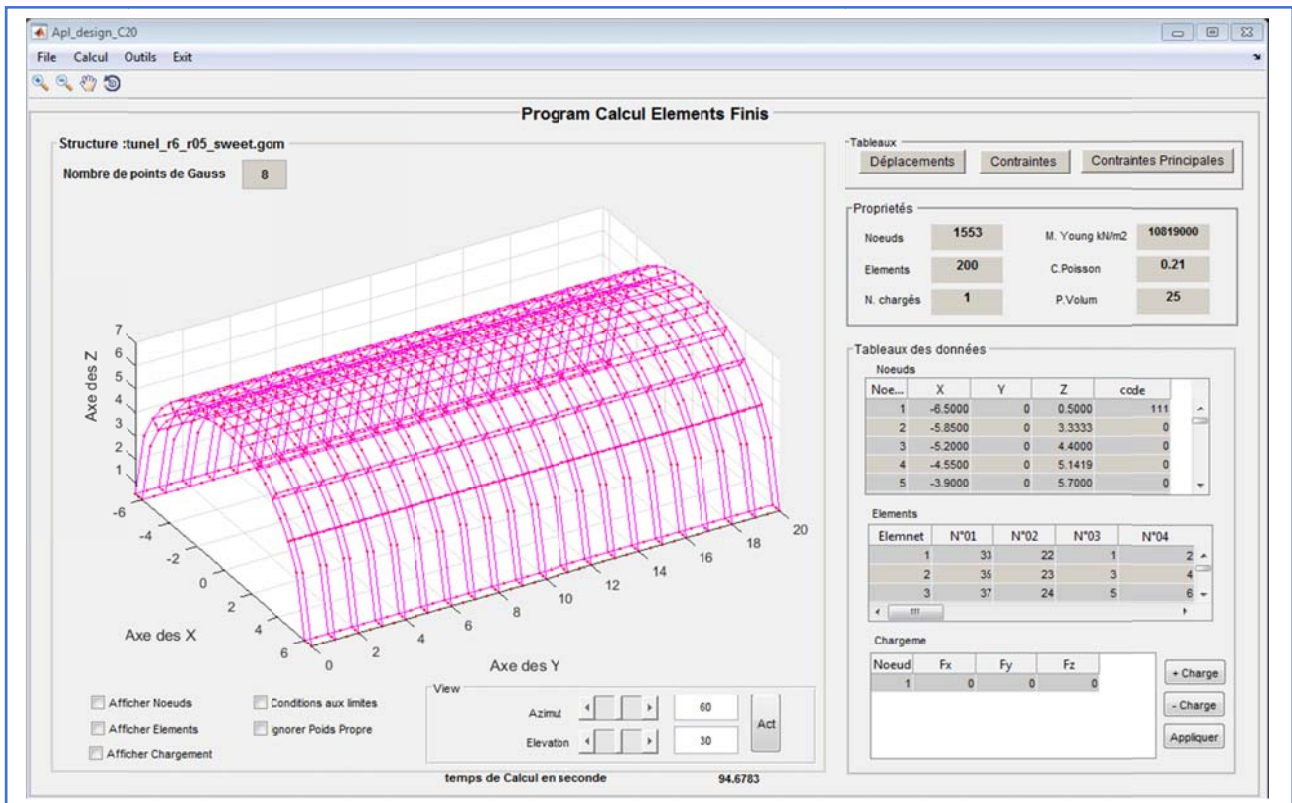


Figure 221 : maillage de la structure

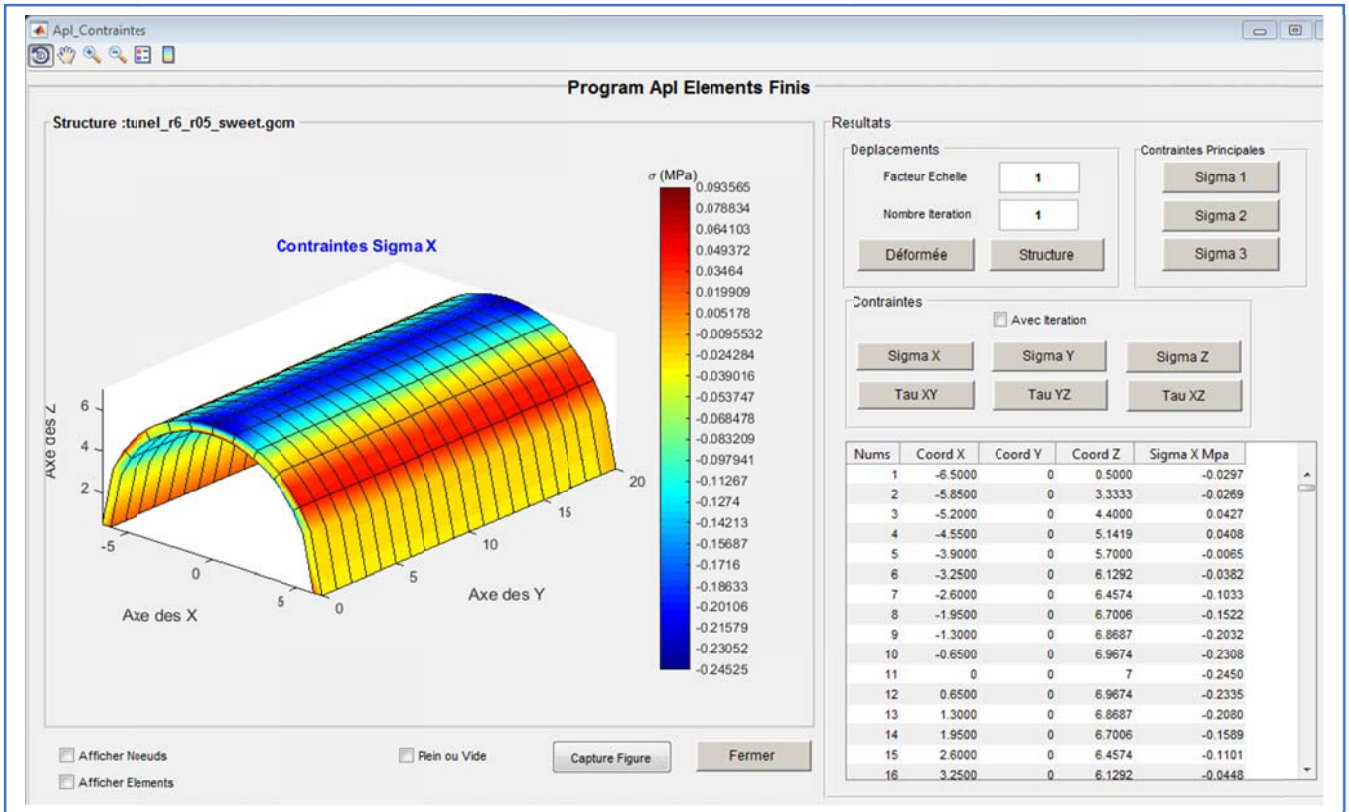


Figure 222 : distribution des contraintes σ_x

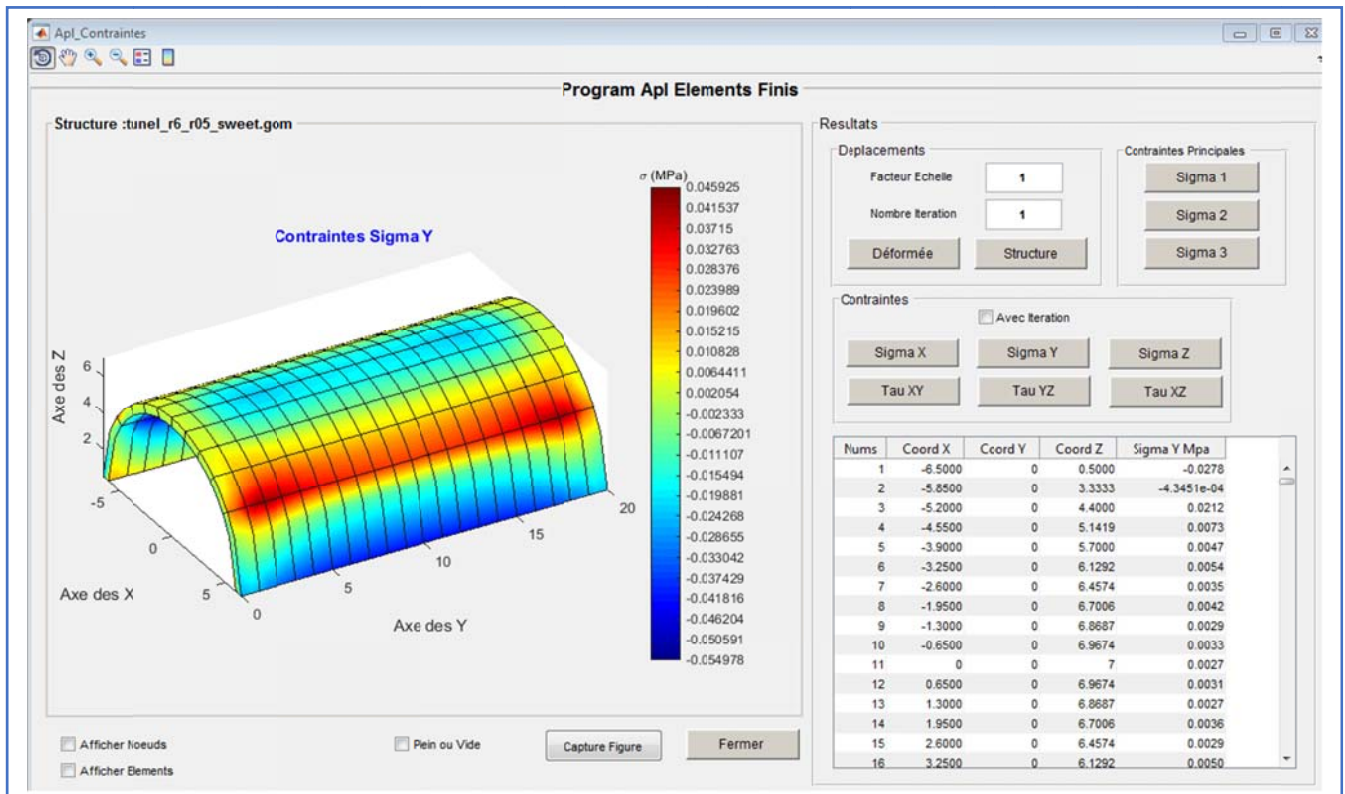


Figure 223 : distribution des contraintes σ_y

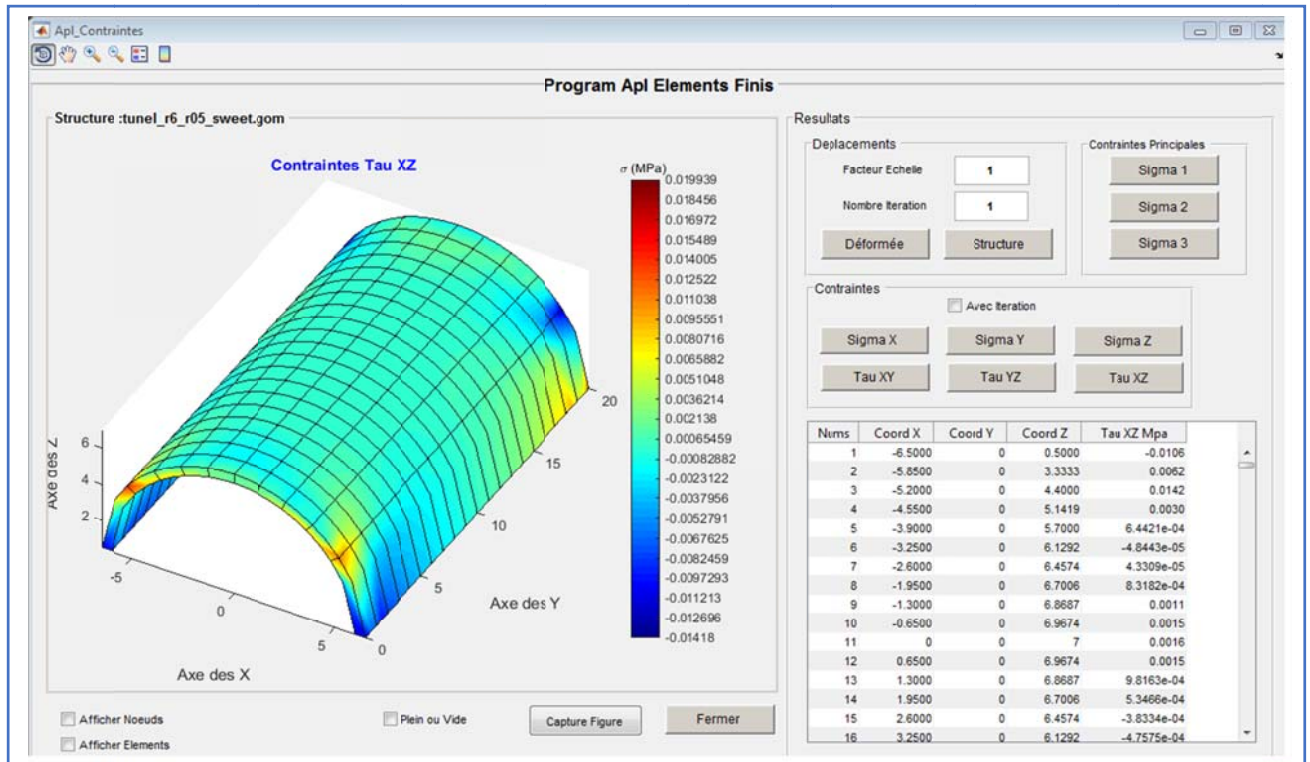


Figure 224 : distribution des contraintes σ_{xy}

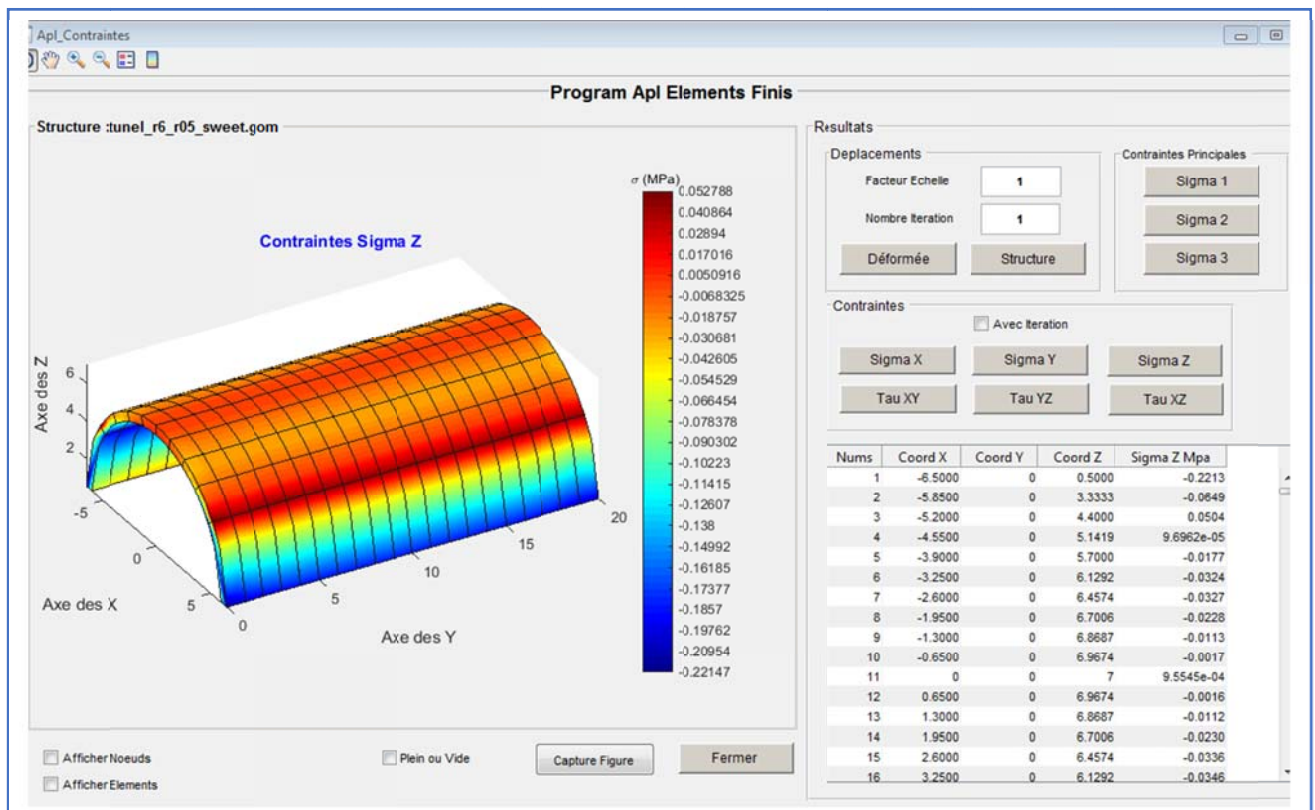


Figure 225 : distribution des contraintes $\sigma_{principale 2}$

5.5.2 Exemple 02

Soit à étudier une Semelle

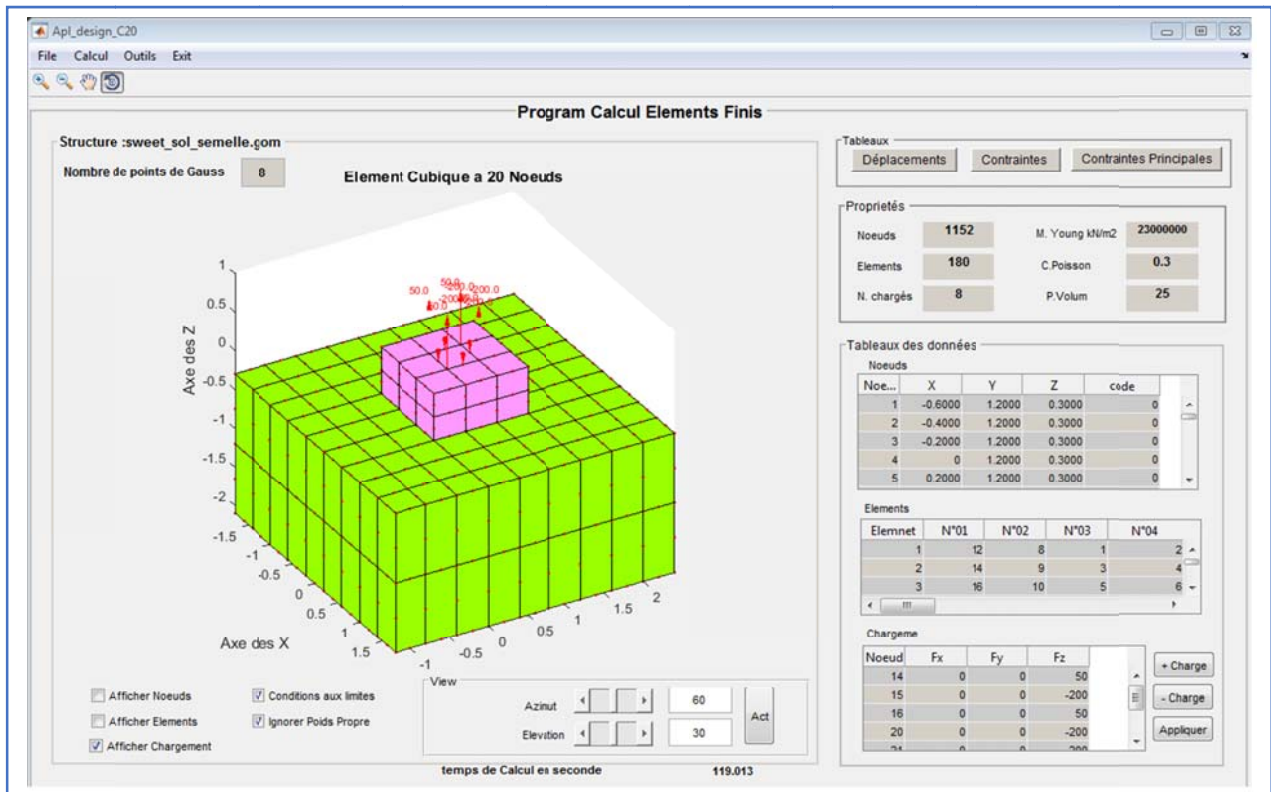


Figure 226 : structure à étudier semelle

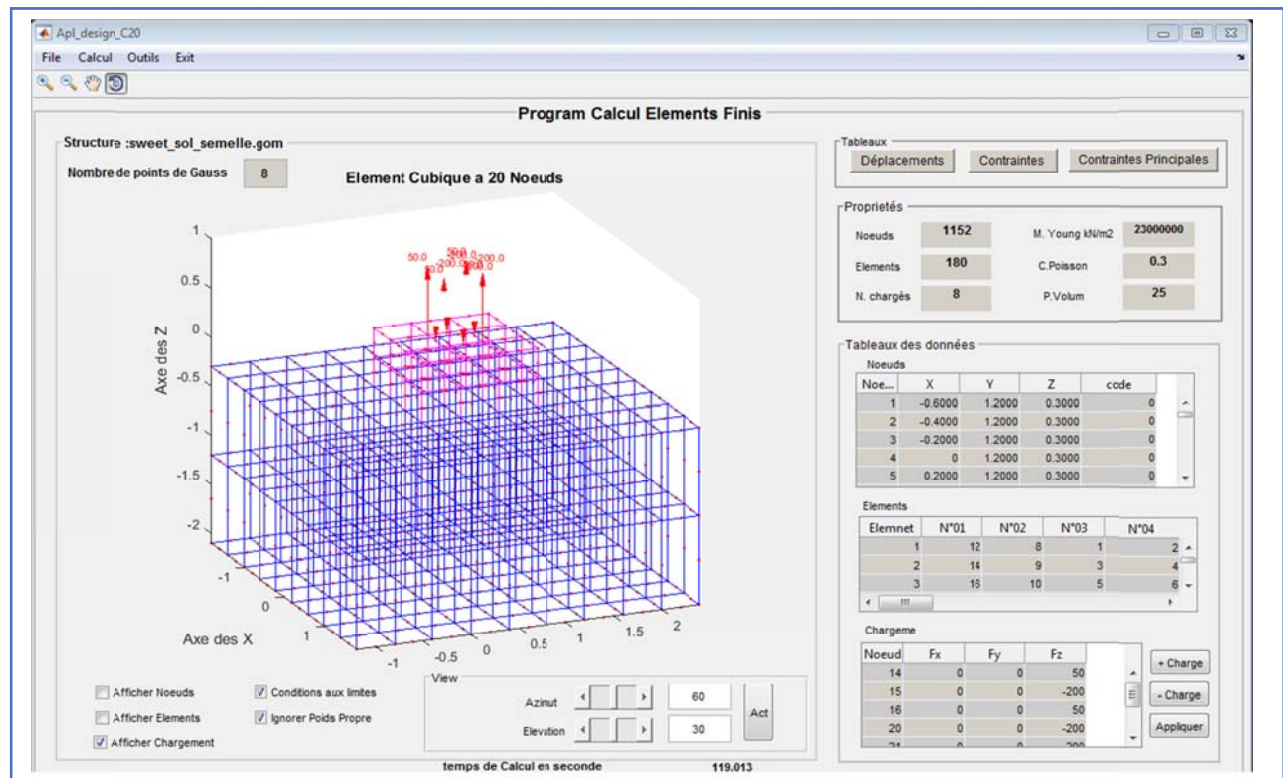


Figure 227 : maillage de la structure avec conditions aux limites

Ainsi les conditions aux limites sont introduites

A savoir encastré a la base et libre dans le sens z sur les cotés

La matrice de rigidité globale est donnée dans la figure 229 avec la demie largeur de bande

Déplacements dans la structure

	Déplacement_Noeuds (cm)			deplacementMax (cm)		
	coord_X	coord_Y	coord_Z	deplc_X	deplc_Y	deplc_Z
1	-0.6000	1.2000	0.3000	3.7669e-04	-3.7669e-04	-1.5177e-04
2	-0.4000	1.2000	0.3000	2.2129e-04	-1.6704e-04	-1.5177e-04
3	-0.2000	1.2000	0.3000	4.4232e-05	-2.2581e-05	-1.5177e-04
4	0	1.2000	0.3000	0	-1.5177e-04	-1.5177e-04
5	0.2000	1.2000	0.3000	-4.4209e-05	-2.2589e-05	-1.5177e-04
6	0.4000	1.2000	0.3000	-2.2127e-04	-1.6704e-04	-1.5177e-04
7	0.6000	1.2000	0.3000	-3.7666e-04	-3.7667e-04	-1.5177e-04
8	-0.6000	1	0.3000	1.6706e-04	-2.2128e-04	-1.5177e-04
9	-0.2000	1	0.3000	3.5222e-04	-4.5033e-04	-1.5177e-04
10	0.2000	1	0.3000	-3.5222e-04	-4.5033e-04	-1.5177e-04
11	0.6000	1	0.3000	-1.6703e-04	-2.2128e-04	-1.5177e-04
12	-0.6000	0.8000	0.3000	2.2601e-05	-4.4219e-05	-1.5177e-04
13	-0.4000	0.8000	0.3000	4.5035e-04	-3.5221e-04	-1.5177e-04
14	-0.2000	0.8000	0.3000	7.2353e-04	-7.2352e-04	-1.5177e-04
15	0	0.8000	0.3000	0	-7.2742e-04	-1.5177e-04
16	0.2000	0.8000	0.3000	-7.2350e-04	-7.2352e-04	-1.5177e-04
17	0.4000	0.8000	0.3000	-4.5032e-04	-3.5221e-04	-1.5177e-04
18	0.6000	0.8000	0.3000	-2.2572e-05	-4.4221e-05	-1.5177e-04
19	-0.6000	0.6000	0.3000	1.5179e-04	0	-1.5177e-04
20	-0.2000	0.6000	0.3000	7.2744e-04	0	-1.5177e-04
21	0.2000	0.6000	0.3000	-7.2741e-04	0	-1.5177e-04
22	0.6000	0.6000	0.3000	-1.5176e-04	0	-1.5177e-04
23	-0.6000	0.4000	0.3000	2.2605e-05	4.4221e-05	-1.5177e-04
24	-0.4000	0.4000	0.3000	4.5035e-04	3.5221e-04	-1.5177e-04

Figure 228 : résultats des déplacements

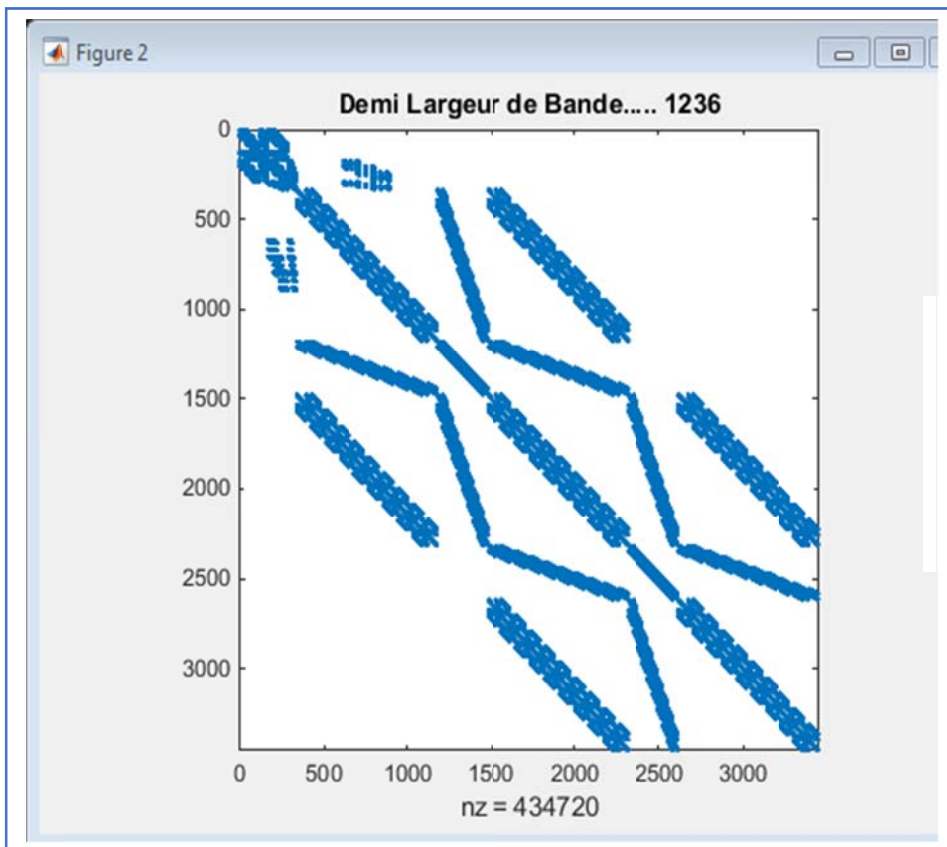


Figure 229 ; matrice de rigidité

Pour mieux visualiser le comportement du sol et la distribution des contraintes, une coupe transversale au milieu de la semelle a été faite

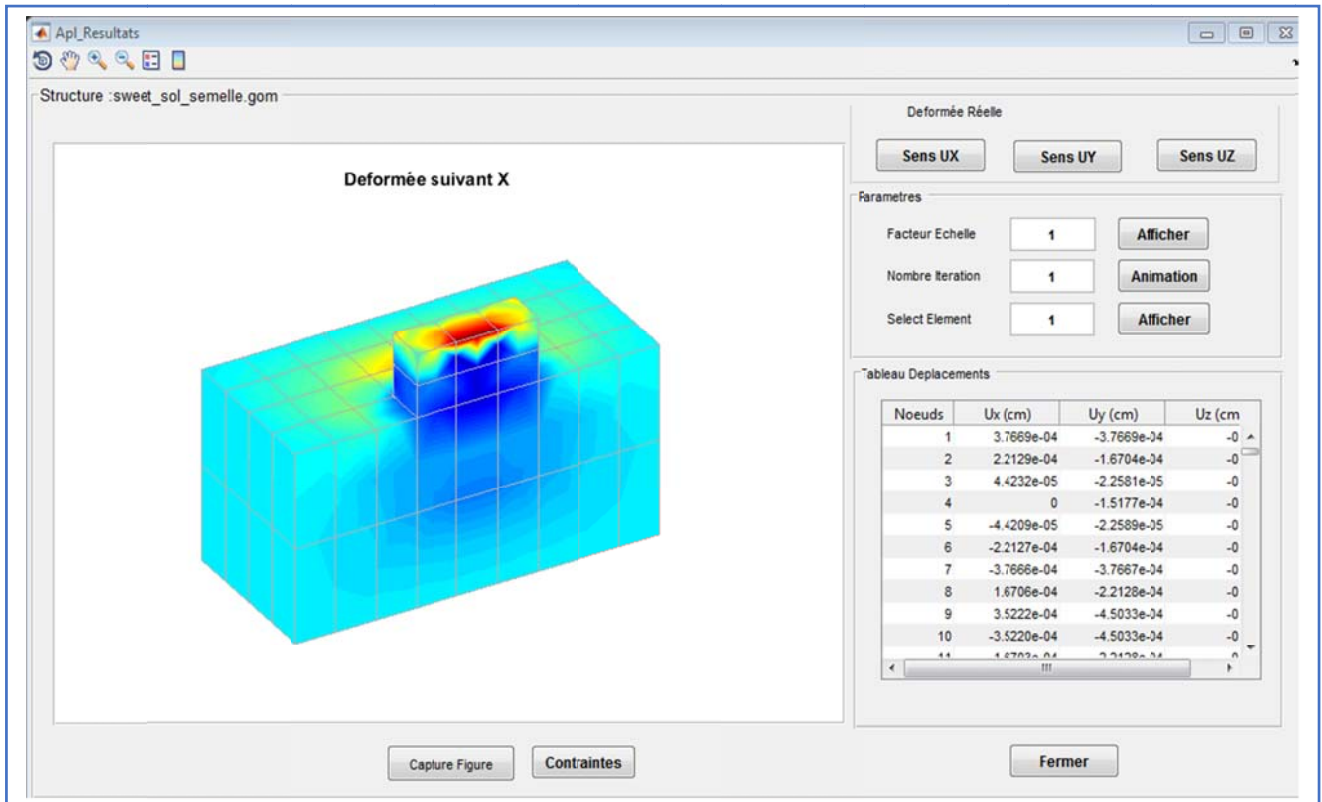


Figure 230 : déformée Ux suivants x

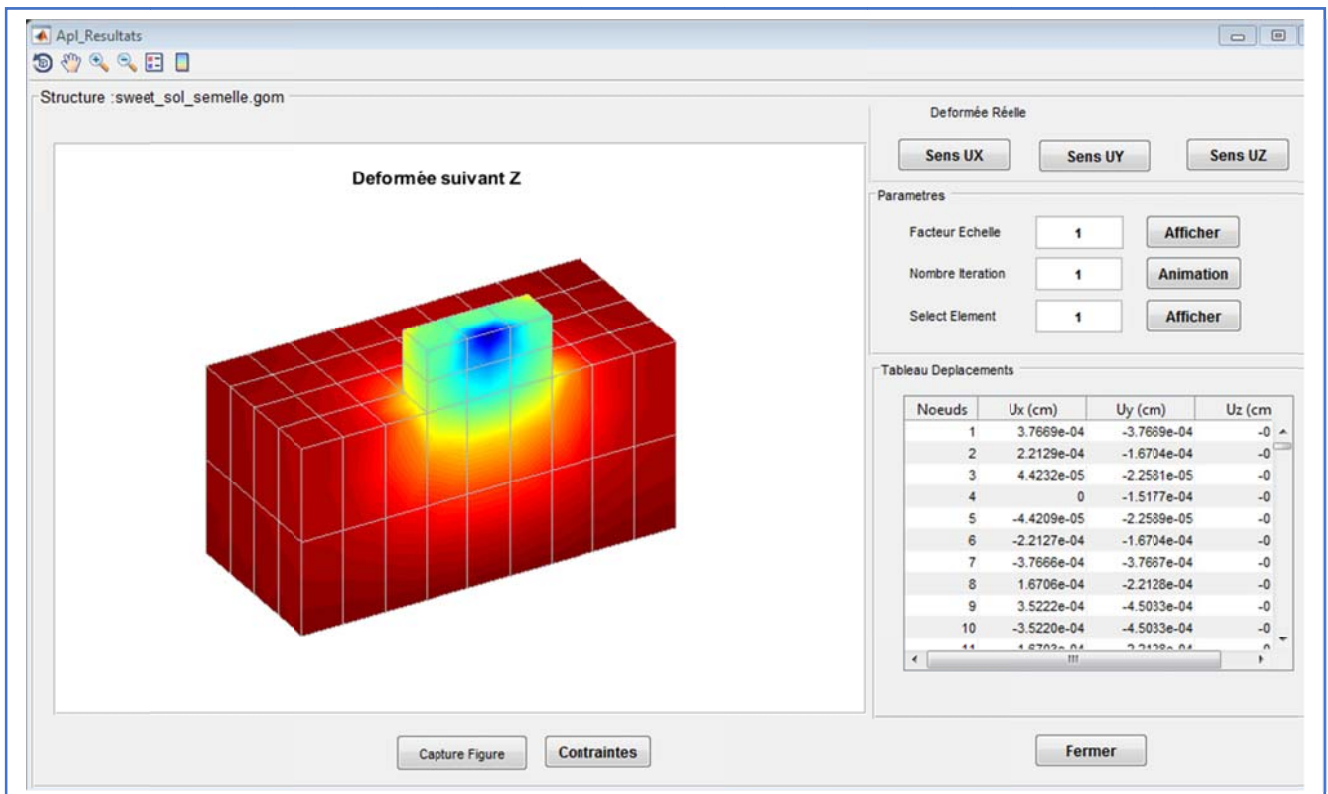


Figure 231 : déformée Uz suivant Z

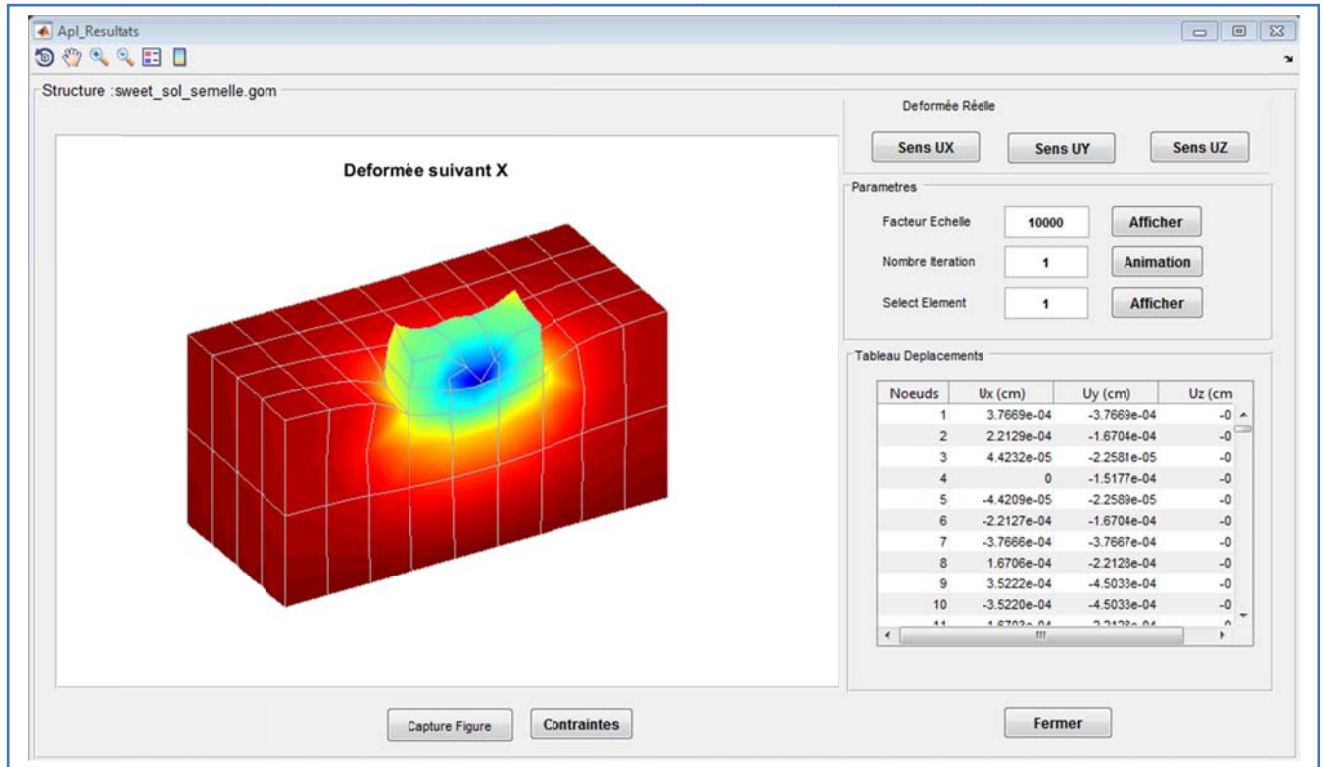


Figure 232 : déformée Ux suivant x

Contraintes dans la structure

	Contrainte_Noeuds (MPa)			Contrainte_PGauss (MPa)		Contrainte_centre_PG (MPa)	
	Sig_X	Sig_Y	Sig_Z	Tau_XY	Tau_XZ	Tau_YZ	
1	3.9597e-06	3.9590e-06	-7.9611e-06	3.3999e-05	-3.5925e-05	3.5926e-05	
2	-9.9129e-06	3.9599e-05	-4.9303e-06	1.8686e-05	-3.2215e-05	-7.6371e-05	
3	-4.3851e-05	8.5348e-05	6.3053e-05	-1.2335e-05	-1.1915e-05	-6.1969e-05	
4	-6.3897e-05	9.5465e-05	1.2801e-04	-4.3928e-10	-6.4420e-10	6.4730e-05	
5	-4.3830e-05	8.5356e-05	6.3066e-05	1.2335e-05	1.1915e-05	-6.1968e-05	
6	-9.9101e-06	3.9601e-05	-4.9306e-06	-1.8685e-05	3.2214e-05	-7.6371e-05	
7	3.9616e-06	3.9615e-06	-7.9682e-06	-3.3998e-05	3.5922e-05	3.5923e-05	
8	3.9600e-05	-9.9126e-06	-4.9264e-06	1.8686e-05	7.6371e-05	3.2215e-05	
9	-1.4740e-04	-7.0245e-05	-1.2019e-04	-1.2736e-05	-8.0167e-05	7.5491e-05	
10	-1.4738e-04	-7.0236e-05	-1.2018e-04	1.2735e-05	8.0167e-05	7.5491e-05	
11	3.9602e-05	-9.9110e-06	-4.9303e-06	-1.8686e-05	-7.6373e-05	3.2215e-05	
12	8.5354e-05	-4.3839e-05	6.3063e-05	-1.2335e-05	6.1969e-05	1.1915e-05	
13	-7.0239e-05	-1.4739e-04	-1.2019e-04	-1.2736e-05	-7.5490e-05	8.0166e-05	
14	-5.7189e-04	-5.7188e-04	-9.9857e-04	-6.6337e-06	-2.5652e-04	2.5652e-04	
15	-9.1792e-04	-8.9281e-04	-0.0017	-1.4178e-10	-5.8691e-10	3.6461e-04	
16	-5.7186e-04	-5.7187e-04	-9.9856e-04	6.6329e-06	2.5652e-04	2.5652e-04	
17	-7.0237e-05	-1.4739e-04	-1.2019e-04	1.2734e-05	7.5490e-05	8.0167e-05	
18	8.5355e-05	-4.3839e-05	6.3063e-05	1.2334e-05	-6.1969e-05	1.1916e-05	
19	9.5466e-05	-6.3897e-05	1.2802e-04	-8.6738e-10	-6.4730e-05	-1.5394e-09	
20	-8.9283e-04	-9.1792e-04	-0.0017	-3.0823e-10	-3.6461e-04	-1.0158e-09	
21	-8.9280e-04	-9.1792e-04	-0.0017	-3.8450e-10	3.6461e-04	7.1403e-10	
22	9.5466e-05	-6.3898e-05	1.2802e-04	-1.9523e-10	6.4729e-05	-3.0060e-10	
23	8.5351e-05	-4.3841e-05	6.3058e-05	1.2335e-05	6.1968e-05	-1.1916e-05	
24	-7.0241e-05	-1.4739e-04	-1.2019e-04	1.2735e-05	-7.5491e-05	-8.0167e-05	
25	-5.7189e-04	-5.7188e-04	-9.9857e-04	6.6335e-06	-2.5652e-04	-2.5652e-04	

Figure 233 : résultats des contraintes

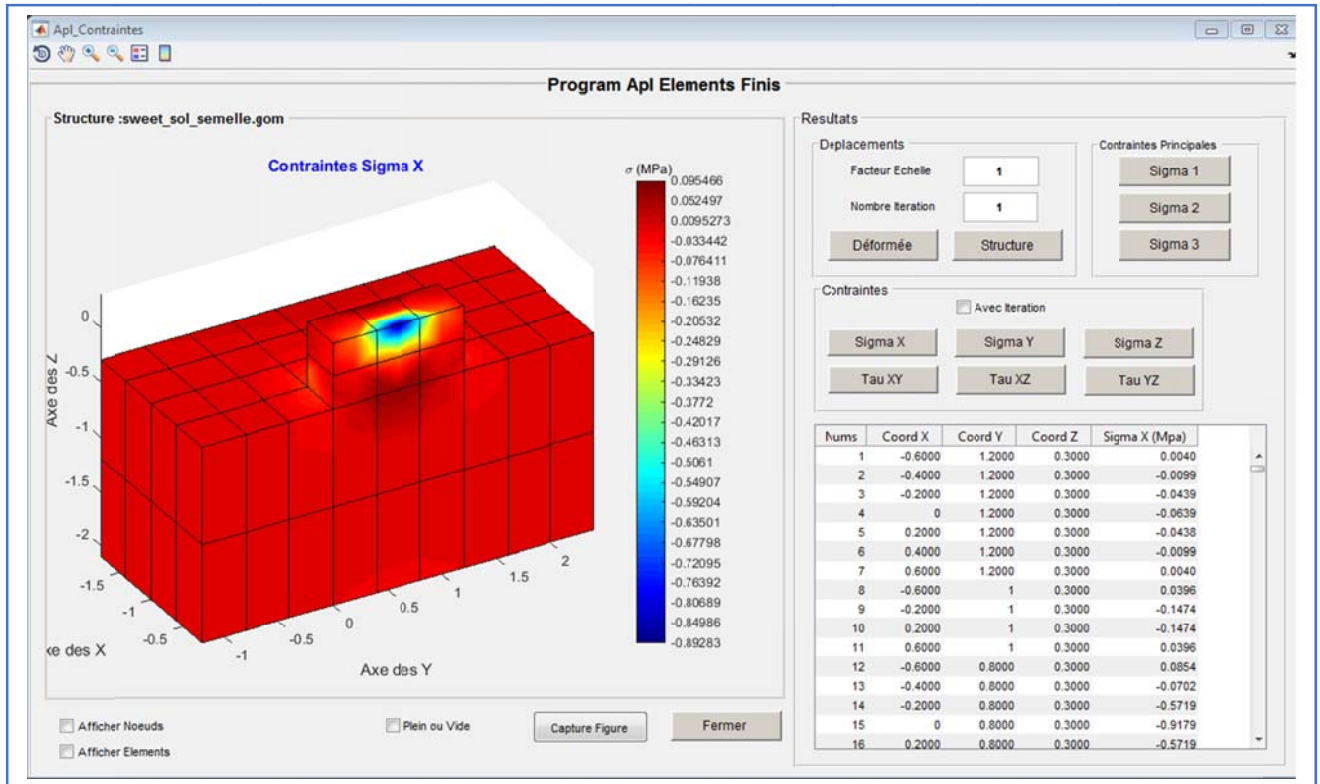


Figure 234 : distribution des contraintes σ_x

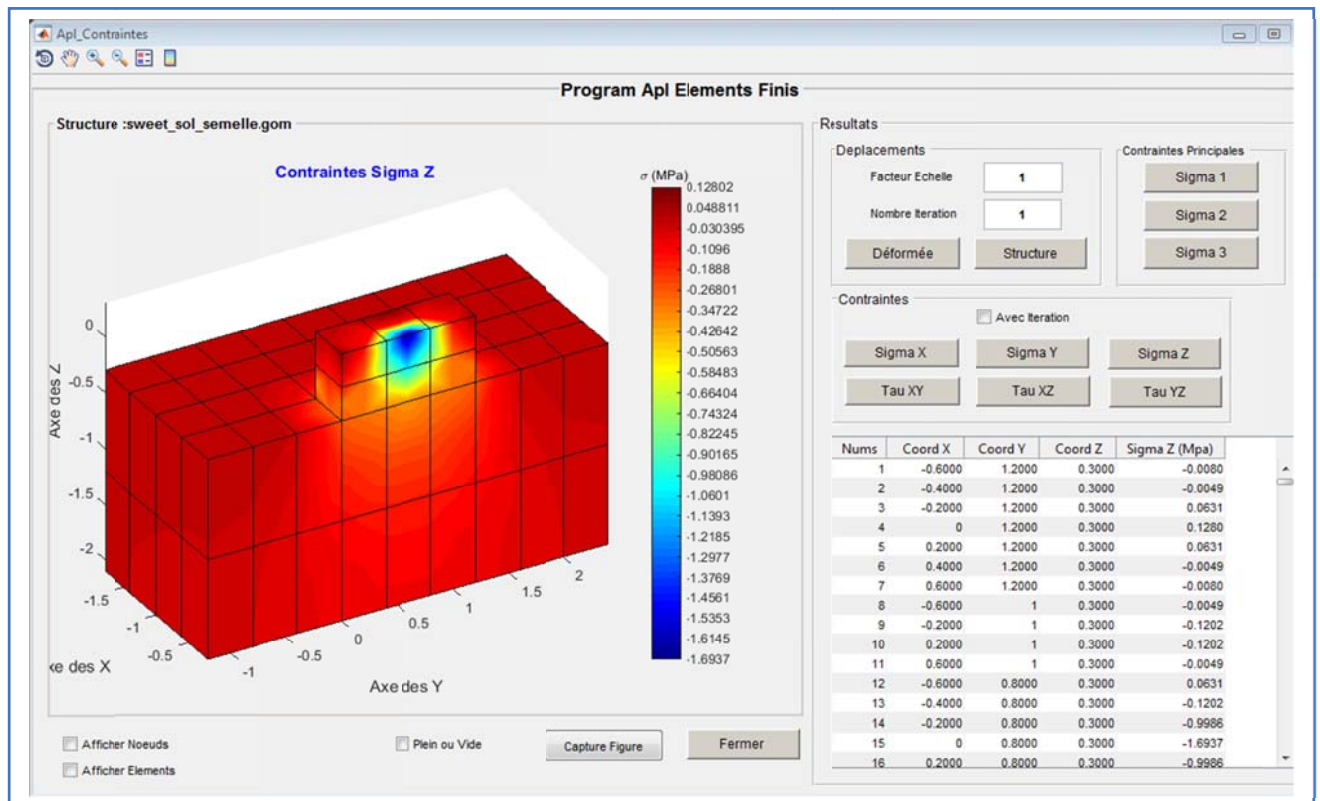


Figure 235 : distribution des contraintes σ_z

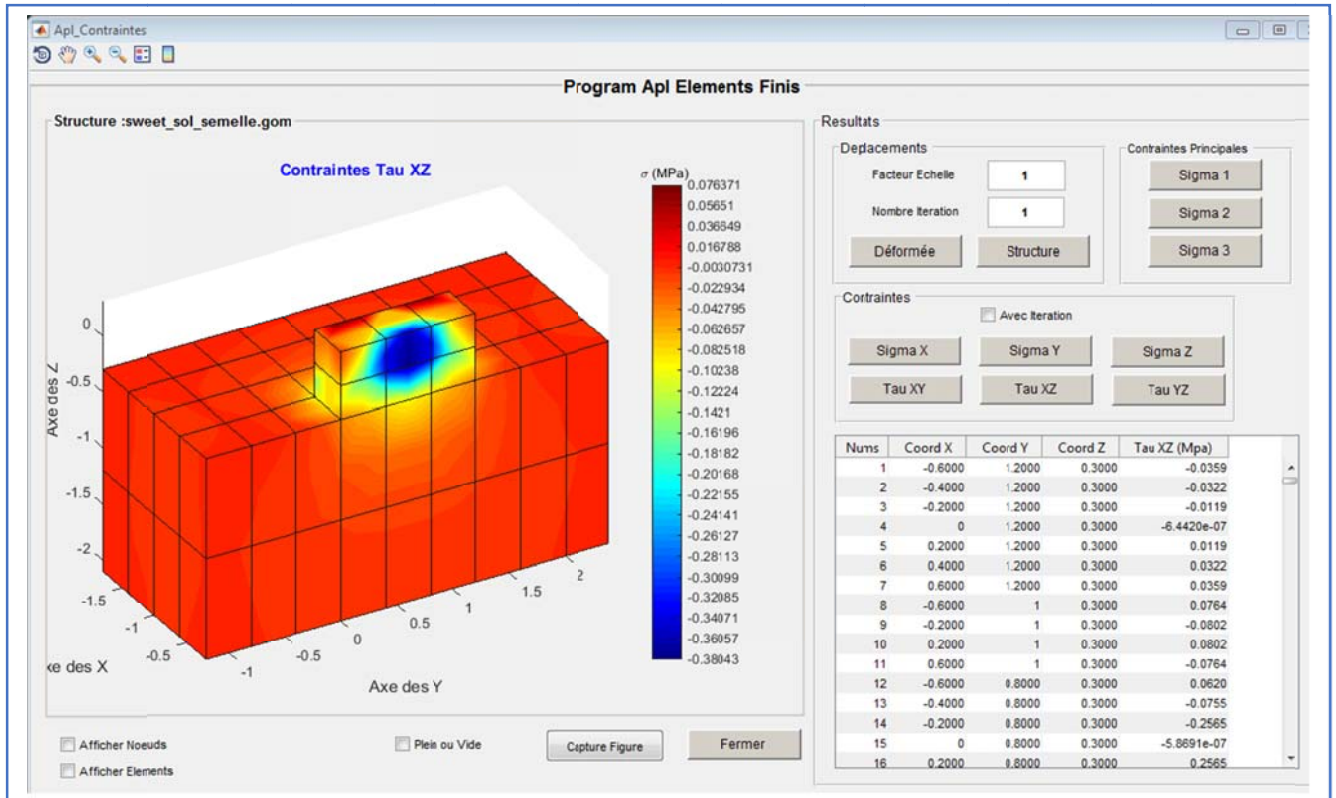


Figure 236 : distribution des contraintes σ_γ

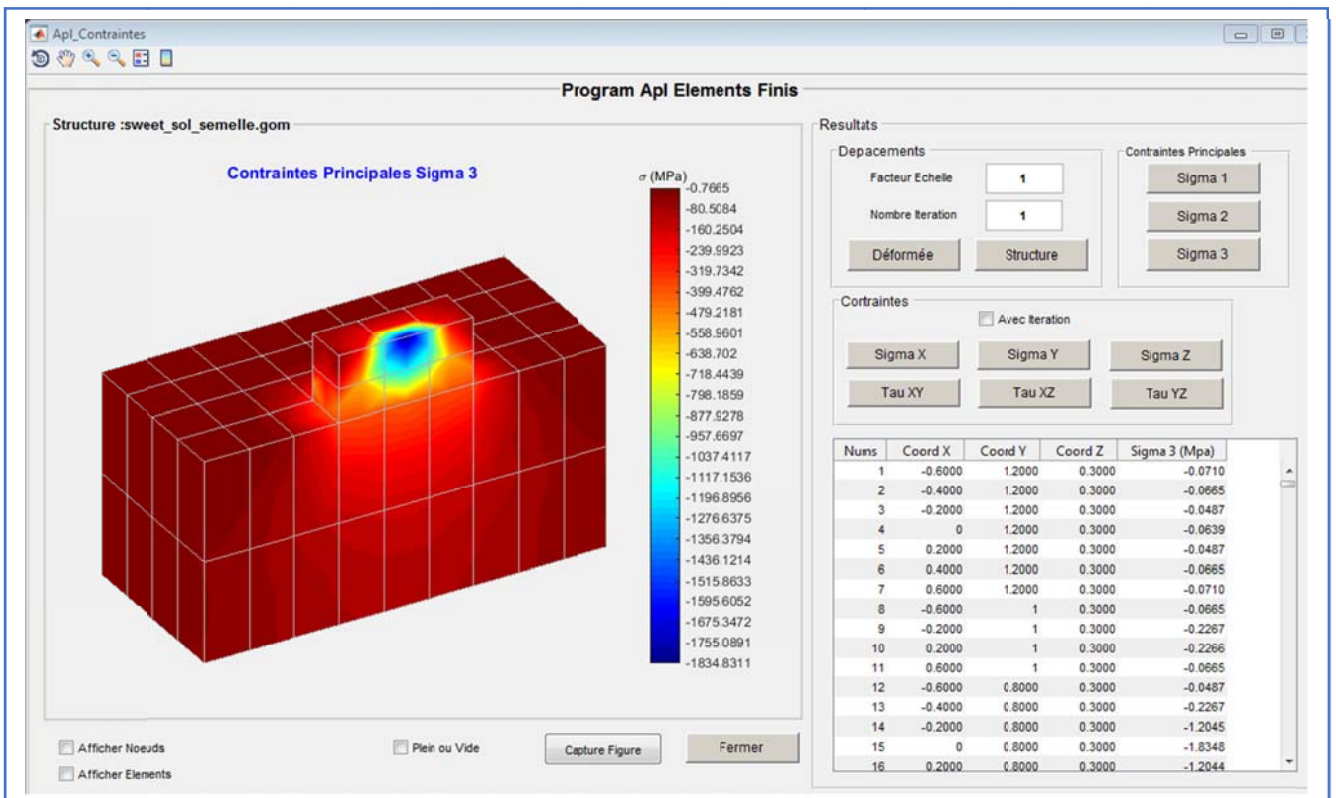


Figure 237 : distribution des contraintes $\sigma_{\text{principale 3}}$

5.5.3 Exemple 03

Soit à étudier un talus en terre sous l'effet d'un chargement surfacique de 600 kN uniformément répartie sur une surface de 8 m x 12 m selon la repartition de la charge equivalente nodale [Ref\[1\]](#) données en annexe B

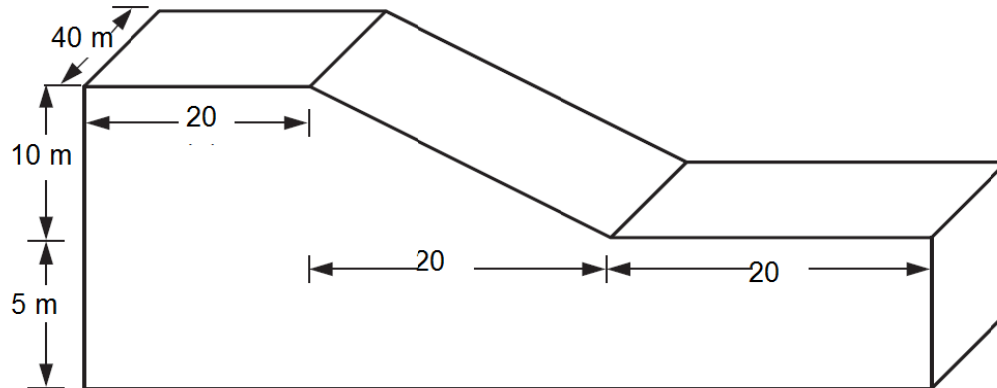


Figure 238 exemple 03

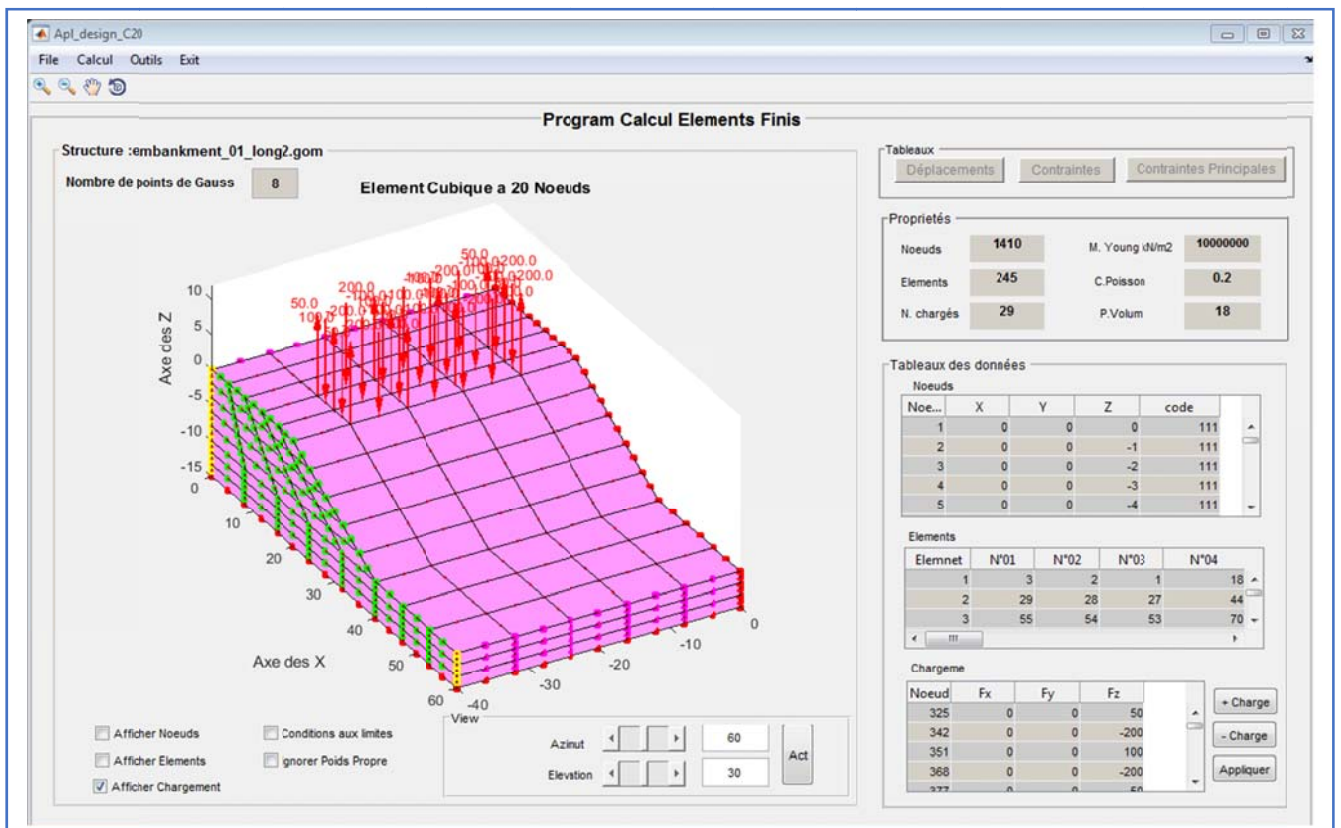


Figure 239 : maillage du terrain avec Apl_mesh hexaedron

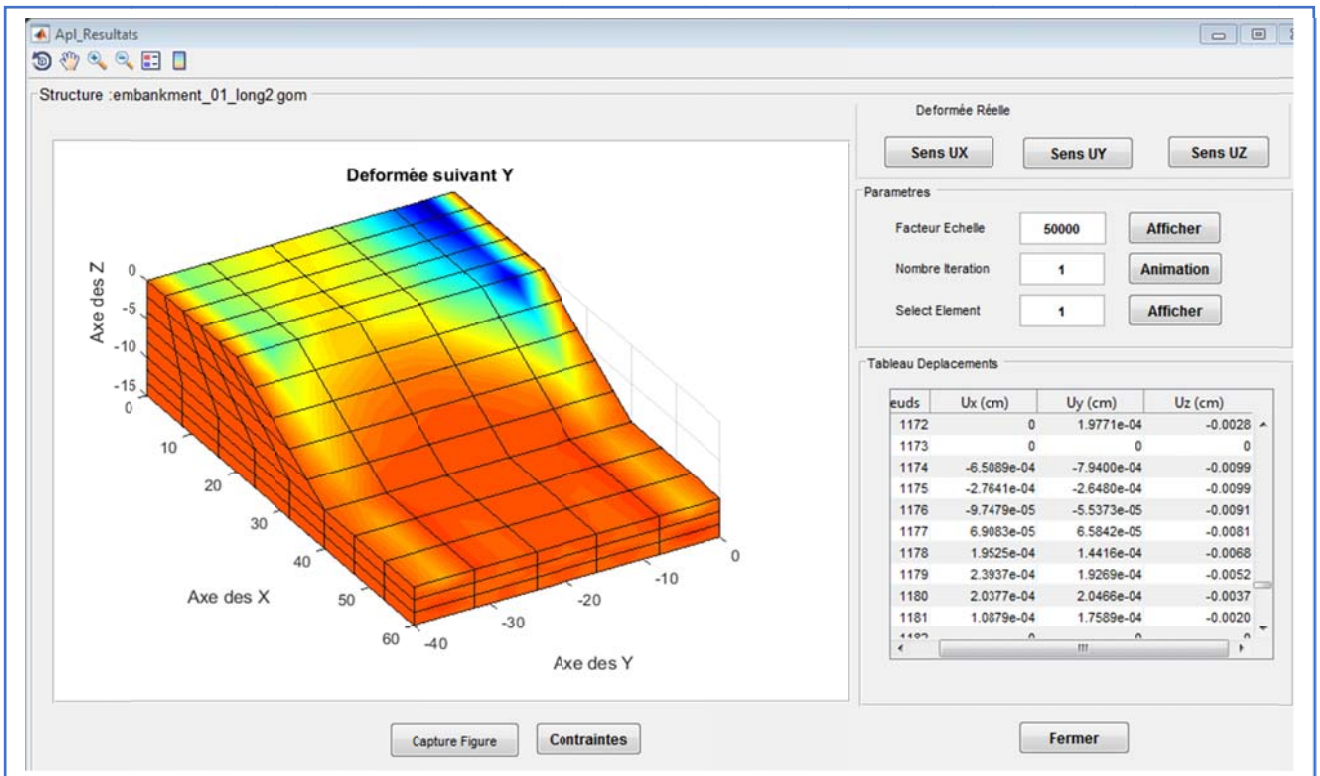


Figure 240 : distribution des déplacement Ux

avec une echelle de 50000

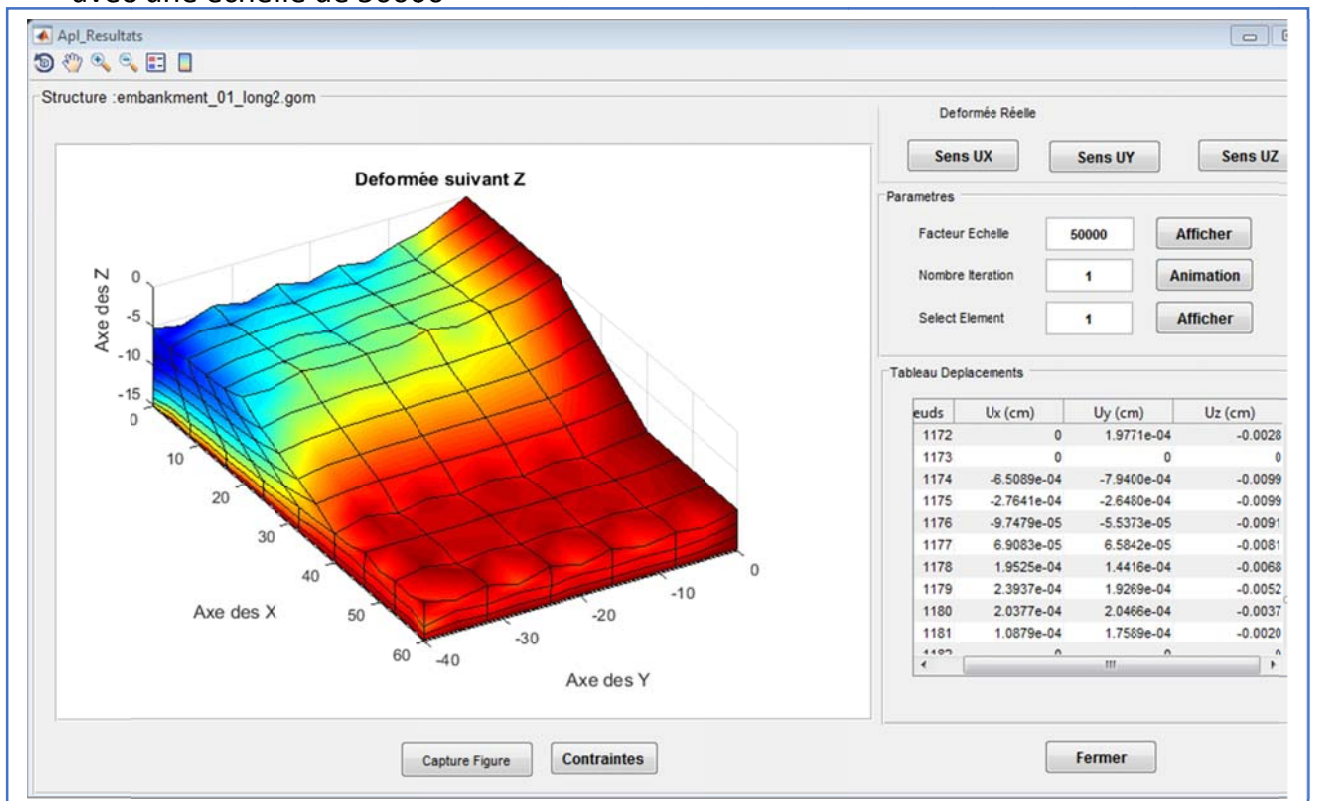


Figure 241 : distribution des déplacement Uz

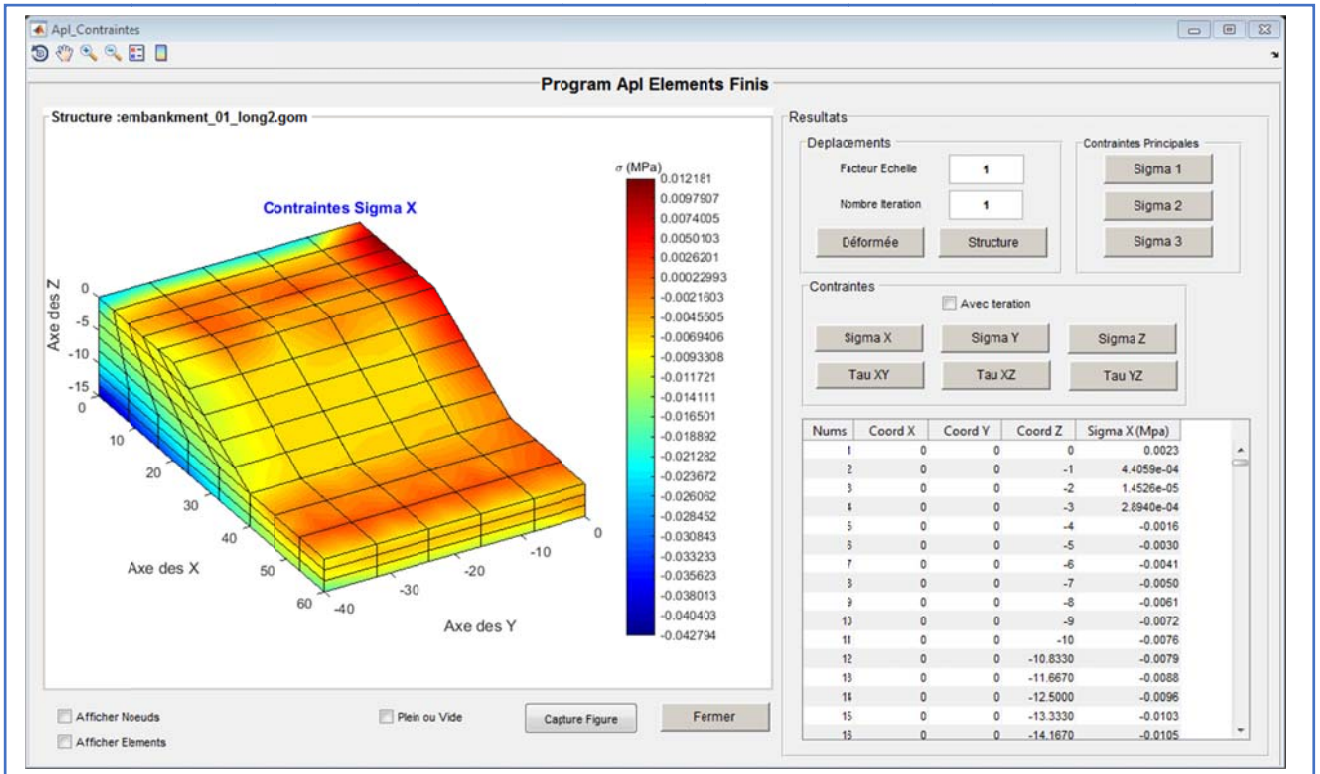


Figure 242 distribution des contraintes σ_x

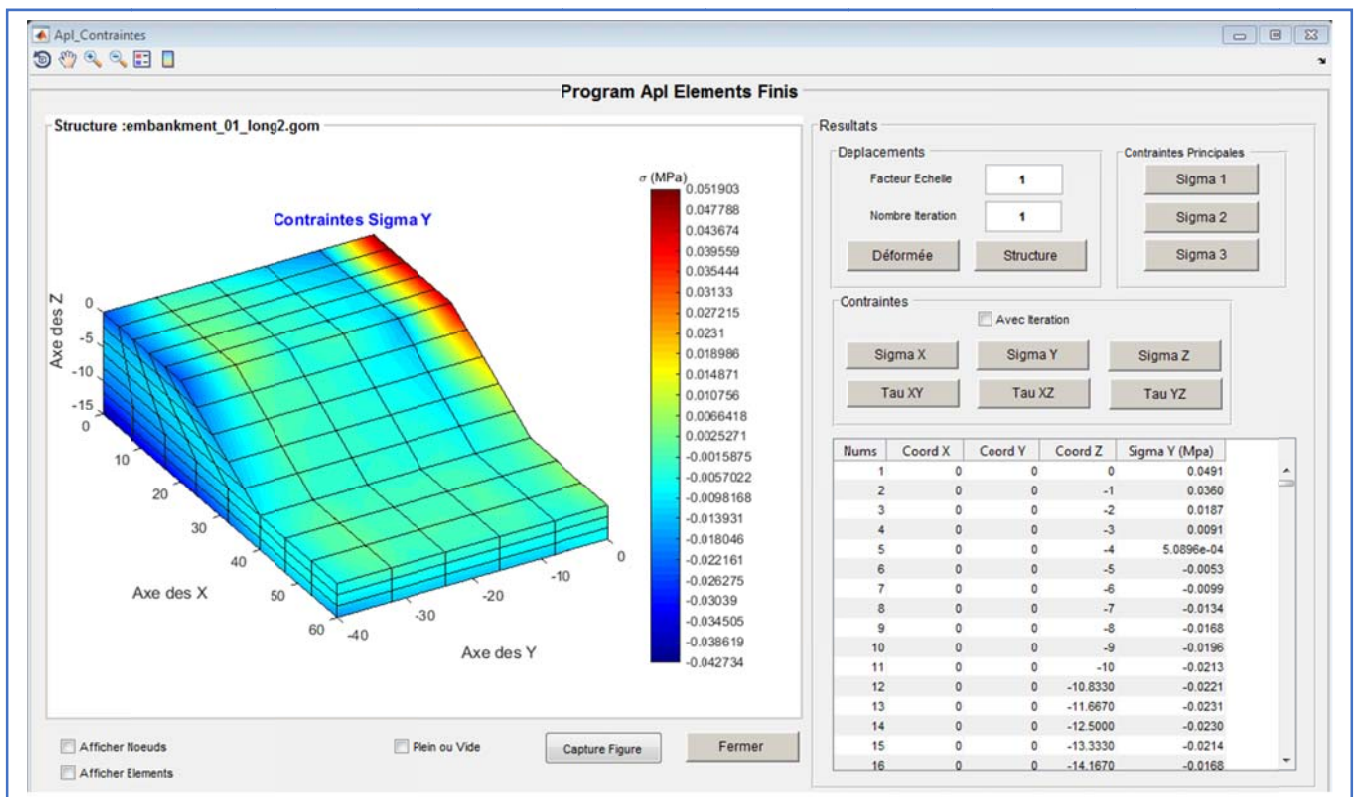


Figure 243: distribution des contraintes σ_y

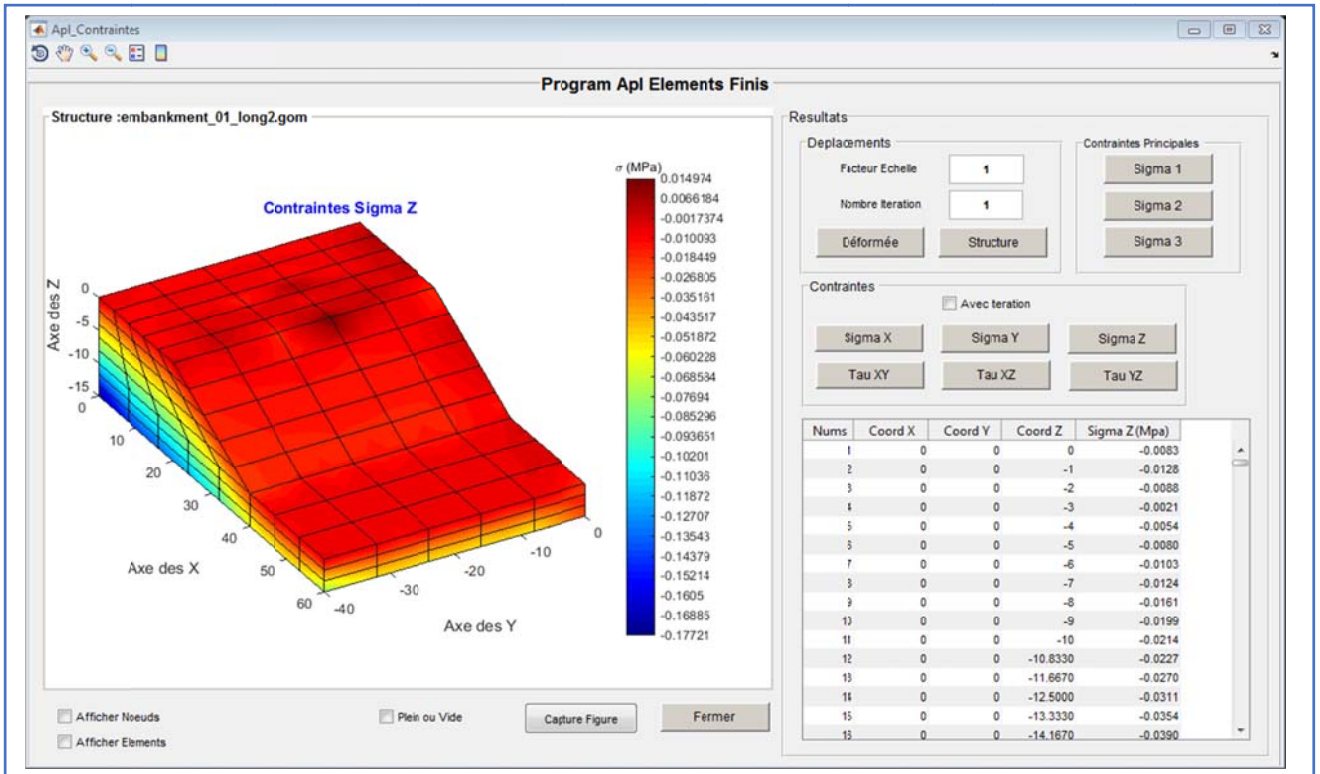


Figure 244 distribution des contraintes σ_z

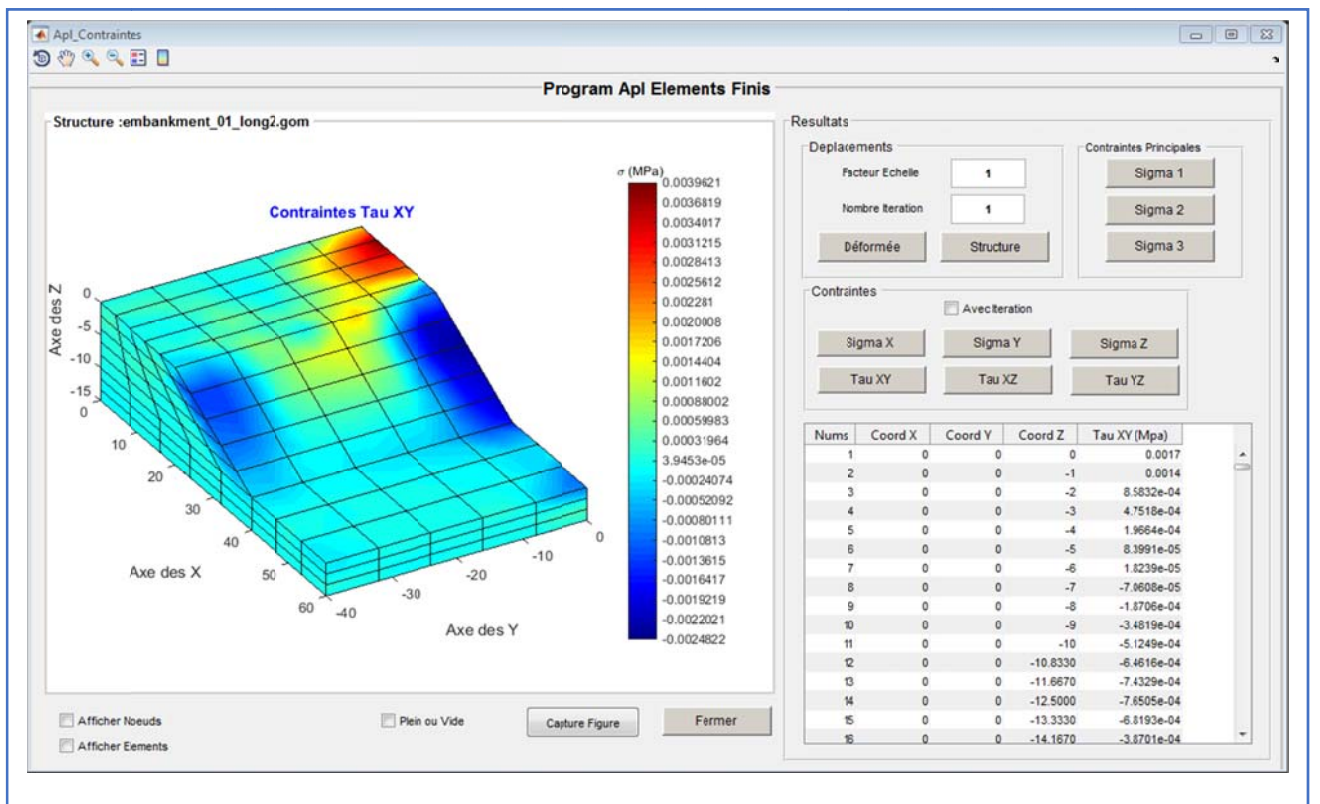


Figure 245: distribution des contraintes τ_{xy}

5.6 Conclusion générale

La validation avec des cas simple ou la RDM est facile à appliquer a démontré la fiabilité du programme **Apl_Design_MARS 1.0** comme il a été constaté dans le chapitre précédant (Validation).

Ainsi des exemples d'applications plus complexes ont été effectués à savoir :

Élément fini Q8 :

- Pilier d'un ouvrage d'art.
- Plaque trouée.
- Tunnel.

Élément fini T6 :

- Plaque métallique

Élément fini cube C8 :

- Dalot 3D sous forme d'arc (béton).
- Poteau métallique H (Acier).

Élément fini Hexa C20 :

- Dalot 3D sous forme arc (béton).
- Semelle isolée (béton, sol).
- Talus en terre (sol).

De manière générale Le but tracé au début de notre travail a été atteint et nous a permis d'ouvrir plusieurs portes dans le domaine linéaire statique et dynamique et de tirer des conclusions concernant la différence entre les types d'éléments, la convergence de certains par rapport à d'autres, ainsi que l'influence de la finesse du maillage élaborés.

Cependant un grand travail reste à faire dans les versions à venir de **Apl_Design_MARS** en vue d'améliorer les performances du logiciel

- Création d'algorithme de maillage pour des structures plus complexes.
- Traitement des problèmes de non-linéarité matérielle et l'implémentation des lois de comportement.
- Les problèmes de non-linéarité géométrique : cas des structures élancées (grands déplacements).

Bibliographies

- [1] I. M. SMITH, D. V. GRIFFITHS, L. MARGETTS - programming the finite element method (2014)
- [2] CAZENAVE, MICHEL - Méthode des éléments finis _ approche pratique en mécanique des structures-Dunod (2013)
- [3] PETER I KATTAN - MATLAB for Beginners_ A Gentle Approach - Revised Edition-CreateSpace Independent Publishing Platform (2010)
- [4] ANTONIO J. M. FERREIRA, NICHOLAS FANTUZZI - MATLAB Codes for Finite Element Analysis_ Solids and Structures (Solid Mechanics and Its Applications (157), Band 157)-Springer (2020)
- [5] SINGIRESU S. RAO - The Finite Element Method in Engineering, Fifth Edition-Elsevier Butterworth Heinemann (2011)
- [6] ZIENKIEWICZ O.C., TAYLOR R.L., The Finite Element Method, Mc Graw-Hill, 1989
- [7] DARYL L. LOGAN _ A First Course in the Finite Element Method
- [8] JACOB FISH, TED BELYTSCHKO - A first course in finite elements-John Wiley & Sons Ltd (2007)
- [9] DARYL L. LOGAN - A First Course in the Finite Element Method by (Fifth Edition)
- [10] SP TIMOSHENKO and JC GOODIER 1970 Theory of Elasticity
- [11] Amar Khennane - Introduction to Finite Element Analysis Using MATLAB® and Abaqus-Boca Raton _ Taylor & Francis (2013)
- [12] KLAUS Jurgen Bathe Edwards L.Wilson Numerical Methods in Finite elemnts Analysis edition 1976
- [13] The Finite Element Method A Practical Course by S. S. Quek, G.R. Liu
- [14] Persson P.O., Mesh size functions for implicit geometries and PDE-based gradiet limiting
- [15] Jacob Fish, Ted Belytschko - A first course in finite elements-John Wiley & Sons Ltd (2007)

[16] Fundamentals of Finite Element Analysis Linear Finite Element Analysis by Ioannis Koutromanos

[17] The Finite Element Method: Its Basis and Fundamentals.

[18] R.L. Taylor, On completeness of shape functions for finite element analysis, Int. J. Numer. Methods Eng. 4 (1972) 17–22.

[19] P. Silvester, Higher order polynomial triangular finite elements for potential problems, Int. J. Eng. Sci. 7 (1969) 849–861.

Annexe A

Sommaire

Apl_lect_dones.....	239
Apl_calcul_barre	240
Apl_calcul_poutres.....	245
Apl_Factor_Cholevsky	250
Apl_newmark_implusion.....	251
Apl_M_rigidite	252
Apl_matrix_B	257
Apl_Fonct_Forme	257
Apl_M_rigidite_trigl	258
Apl_Fct_Form_trigl.....	261
Apl_Tetra_free	262
Apl_matrice_D	265
Apl_Point_Gauss_tetra.....	266
Apl_BJdet_tetra	267
Cprinc1	267
Apl_cub_8N.....	268
Apl_Fonct_Forme_cub8.....	272
Apl_Cube_20N_free	273
Apl_Fonct_Forme_cub20.....	277
Apl_Points_Gauss.....	278

Apl_lect_dones

```

function [] = Apl_lect_dones ()
global n GMXY GNN GNE GNMV GME x y fdg EM TGNNCL nc TCXY  AXYNC GNNCL GNNM
global encast blocky blockx moduleYoung masseVolum coefPoisson filename
fdg=fopen(filename,'r');
% lecture ds données
GNN=fscanf(fdg,'%5d',1); %nombre de noeud
GNE=fscanf(fdg,'%5d',1); % nombre d'elemnt
GNMV=fscanf(fdg,'%5d',1); %nombre de materiau
GYXM=fscanf(fdg,'%f',[4,GNN]);
GMXY=transpose(GYXM);
EM=fscanf(fdg,'%f',[13,GNE]);
GME=EM';
GNNM=fscanf(fdg,'%5d',1); %nombre de materiaux
moduleYoung =fscanf(fdg,'%f',1); %valeur du module de Young
masseVolum =fscanf(fdg,'%f',1); %valeur de ro masse volumique
coefPoisson =fscanf(fdg,'%f',1); %valeur coefficient de poisson
GNNCL=fscanf(fdg,'%5d',1); % noueds aux condition limite
Fnod=fopen('nodes.out','w');
fprintf(Fnod,'%d %d %d %d %d %d %d %d %d %d %d %d %d %f\n ',GME');
% lecture des conditions aux limites
CNCL=fscanf(fdg,'%f',[3,GNNCL]);
TGNNCL=transpose(CNCL);
n=GMXY(:,1);
x=GMXY(:,2);
y=GMXY(:,3);
nc=TGNNCL(:,1); %liste des noeuds chargés
TCXY=[TGNNCL(:,2) TGNNCL(:,3)];
for I=1:GNNCL
    nuimc = nc(I,1);
    vxnc = GMXY(nuimc,2);
    vync = GMXY(nuimc,3);
    AXYNC(I,:)= [nuimc, vxnc, vync];
end
AXYNC=AXYNC;
AXYN=[AXYNC TCXY];
NCOD=GMXY(:,4);
length(n);
encast=zeros(length(n),3);
blockx=zeros(length(n),3);
blocky=zeros(length(n),3);
for i=1:GNN
    vncd = NCOD(i,1);
    nncd = n(i,1);
    if vncd == 11
        nnncd=n(nncd,1);
        xcod = GMXY(nncd,2);
        ycod = GMXY(nncd,3);
        encast(i,:)= [nnncd xcod ycod];
    end
    if vncd == 01
        nnncd=n(nncd,1);
        xcod = GMXY(nncd,2);
        ycod = GMXY(nncd,3);
        blocky(i,:)= [nnncd xcod ycod];
    end
    if vncd == 10
        nnncd=n(nncd,1);
        xcod = GMXY(nncd,2);
        ycod = GMXY(nncd,3);
        blockx(i,:)= [nnncd xcod ycod];
    end
end
end
Apl_condition_limit () % affichage conditions aux limites
end

```

Apl_calcul_barre

```

function []=Apl_calcul_barre()
global NE EN NN CN TNF NF MDD DEP VF KG VFG VVVT MG KGddl vddl facVF
MRHV MVdep MNIN INCN
TNCL=TNF;
% Matrice de rigidité
KG=zeros(2*NN,2*NN);
MG=zeros(2*NN,2*NN);
E2N=[EN(:,2) EN(:,3)];
VFVT=zeros(2*NN,1); % vecteur forces total

for P=1:NE
E=EN(P,6);
RO=EN(P,7);
S=EN(P,4)*EN(P,5); % section de la barre cas isotrope
N1=EN(P,2);
N2=EN(P,3);
dx=(CN(EN(P,3),2)-CN(EN(P,2),2));
dy=(CN(EN(P,3),3)-CN(EN(P,2),3));
L=sqrt(dx^2+dy^2)
% prise en compte des forces de volume
Fvro=[0
      -S*RO*L/2
      0
      -S*RO*L/2]
VFVT(2*N1-1)= VFVT(2*N1-1)+Fvro(1);
VFVT(2*N1)= VFVT(2*N1)+Fvro(2);
VFVT(2*N2-1)= VFVT(2*N2-1)+Fvro(3);
VFVT(2*N2)= VFVT(2*N2)+Fvro(4);
KE=[ (dx/L)^2          (dx/L)*(dy/L)    - (dx/L)^2          - (dx/L)*(dy/L)
      (dx/L)*(dy/L)    (dy/L)^2        - (dx/L)*(dy/L)    - (dy/L)^2
      - (dx/L)^2       - (dx/L)*(dy/L)    (dx/L)^2          (dx/L)*(dy/L)
      - (dx/L)*(dy/L) - (dy/L)^2        (dx/L)*(dy/L)    (dy/L)^2];
Re=[ dx/L   dy/L   0   0
      -dy/L  dx/L   0   0
      0      0     dx/L dy/L
      0      0     -dy/L dx/L]

K = (E*S/L) * [ 1 0 -1 0          % matrice de rigidité dans le repère de la barre
               0 0 0 0
               -1 0 1 0
               0 0 0 0]

M = (RO*S*L/6) * [ 2 0 1 0          % matrice masse dans le repère lié à la barre
                  0 2 0 1
                  1 0 2 0
                  0 1 0 2]
Mec = 0.5*RO*S*L*eye(4)
b=[ dx; dy ];
Ll=sqrt(b'*b);
Kle=E*S/Ll*[ 1 -1;
             -1 1];
n=b'/Ll;  Gl=[ n          zeros(size(n));
              zeros(size(n)) n    ];
Ke=Gl'*Kle*Gl;
Me=Re'*Mec*Re ;
ESL=E*S/L;
KE=KE*ESL;
% Assemblage de la matrice de rigidité
KG(2*N1-1,2*N1-1)= KG(2*N1-1,2*N1-1)+ KE(1,1);
KG(2*N1-1,2*N1) = KG(2*N1-1,2*N1) + KE(1,2);
KG(2*N1-1,2*N2-1)= KG(2*N1-1,2*N2-1)+KE(1,3);
KG(2*N1-1,2*N2) = KG(2*N1-1,2*N2)+KE(1,4);

```



```

KG(2*N1,2*N1-1)= KG(2*N1,2*N1-1)+ KE(2,1);
KG(2*N1,2*N1) = KG(2*N1,2*N1) + KE(2,2);
KG(2*N1,2*N2-1)=KG(2*N1,2*N2-1)+KE(2,3);
KG(2*N1,2*N2)=KG(2*N1,2*N2)+KE(2,4);
KG(2*N2-1,2*N1-1)= KG(2*N2-1,2*N1-1)+ KE(3,1);
KG(2*N2-1,2*N1) = KG(2*N2-1,2*N1) + KE(3,2);
KG(2*N2-1,2*N2-1)= KG(2*N2-1,2*N2-1) + KE(3,3);
KG(2*N2-1,2*N2) = KG(2*N2-1,2*N2) + KE(3,4);
KG(2*N2,2*N1-1)= KG(2*N2,2*N1-1)+KE(4,1);
KG(2*N2,2*N1) = KG(2*N2,2*N1)+KE(4,2);
KG(2*N2,2*N2-1)=KG(2*N2,2*N2-1)+KE(4,3);
KG(2*N2,2*N2) =KG(2*N2,2*N2) +KE(4,4);
% Assemblage de la matrice de masse
MG(2*N1-1,2*N1-1)= MG(2*N1-1,2*N1-1)+ Me(1,1);
MG(2*N1-1,2*N1) = MG(2*N1-1,2*N1) + Me(1,2);
MG(2*N1-1,2*N2-1)= MG(2*N1-1,2*N2-1)+Me(1,3);
MG(2*N1-1,2*N2) = MG(2*N1-1,2*N2)+Me(1,4);
MG(2*N1,2*N1-1)= MG(2*N1,2*N1-1)+ Me(2,1);
MG(2*N1,2*N1) = MG(2*N1,2*N1) + Me(2,2);
MG(2*N1,2*N2-1)=MG(2*N1,2*N2-1)+Me(2,3);
MG(2*N1,2*N2)=MG(2*N1,2*N2)+Me(2,4);
MG(2*N2-1,2*N1-1)= MG(2*N2-1,2*N1-1)+ Me(3,1);
MG(2*N2-1,2*N1) = MG(2*N2-1,2*N1) + Me(3,2);
MG(2*N2-1,2*N2-1)= MG(2*N2-1,2*N2-1) + Me(3,3);
MG(2*N2-1,2*N2) = MG(2*N2-1,2*N2) + Me(3,4);
MG(2*N2,2*N1-1)= MG(2*N2,2*N1-1)+Me(4,1);
MG(2*N2,2*N1) = MG(2*N2,2*N1)+Me(4,2);
MG(2*N2,2*N2-1)=MG(2*N2,2*N2-1)+Me(4,3);
MG(2*N2,2*N2) =MG(2*N2,2*N2) +Me(4,4);
end
KGint=KG;
ad=1; %condition aux limite
cpr=0;
for P=1 :NN
adr= CN(P,1);
if CN(P,4)==11
cpr=cpr+1;
KG(2*adr-1,2*adr-1)=KG(2*adr-1,2*adr-1)*1.0e+30;
KG(2*adr,2*adr)=KG(2*adr,2*adr)*10^25;
vddl(ad)=2*CN(P,1)-1;
vddl(ad+1)=2*CN(P,1);
ad=ad+2;
%sprintf('Encastrement')
end
if CN(P,4)==10
cpr=cpr+1;
KG(2*adr-1,2*adr-1)=KG(2*adr-1,2*adr-1)*1.0e+30;
%sprintf('bloker sur x');
vddl(ad)=2*CN(P,1)-1;
ad=ad+1;
end
if CN(P,4)==1
cpr=cpr+1;
KG(2*adr,2*adr)=KG(2*adr,2*adr)*1.0e+30;
%sprintf('bloker sur y');
vddl(ad)=2*CN(P,1);
ad=ad+1;
end; end
KGddl=KG %fprintf('%s','Matrice KG apres conditions au limites');
VF=zeros(NN*2,1); % vecteur de Forces
for ii=1:NF
NC=TNCL(ii,1);
FX=TNCL(ii,2);
FY=TNCL(ii,3);
VF(2*NC-1)=FX;
VF(2*NC)=FY;
end
facVF=1

```

```

% VFVG vecteur force total pris en compte du poids propres VFVT+ charge VF
VF=facVF*VFVT+VF
DEP=KGddl\VF;
n=length(DEP);
  RHV=KGint*DEP;
  DEPG=KG\RHV;
  % determination des effort interne
  % dans le repere locale
  NIN = zeros(NE,5); % Matrice des effort interne
  CONT = zeros(NE,5); % Matrice des contraintes
for P=1:NE
E=EN(P,6);
S=EN(P,4)*EN(P,5);
N1=EN(P,2);
N2=EN(P,3);
dx=(CN(EN(P,3),2)-CN(EN(P,2),2));
dy=(CN(EN(P,3),3)-CN(EN(P,2),3));
L=sqrt(dx^2+dy^2);
  RO=EN(P,7); %poids volumique
  % matrice de rigidite de chaque element
  % repere locale
ESL=E*S/L;
ke=[ 1 0 -1 0
     0 0 0 0
    -1 0 1 0
     0 0 0 0]*ESL;
Re=[ dx/L   dy/L   0   0
     -dy/L  dx/L   0   0
        0    0   dx/L  dy/L
        0    0  -dy/L  dx/L];
Dpeg=[DEPG(2*N1-1,1)
      DEPG(2*N1,1)
      DEPG(2*N2-1,1)
      DEPG(2*N2,1)];
  % déplacement dans le repere locale
qeg=Re*Dpeg ; % déplacement dans le repere locale

  % effort interne dans le repere locale
Nintg =ke*qeg ;
Dpe=[DEP(2*N1-1,1)
     DEP(2*N1,1)
     DEP(2*N2-1,1)
     DEP(2*N2,1)];
  % adresse des déplacement
sctr=[2*N1-1 2*N1 2*N2-1 2*N2];
DEP(sctr);
qe=Re*Dpe; % déplacement dans le repere locale
% effort interne dans le repere locale
Nint=ke*qe ;
  % Matrice des effort interne
NIN(P,:) = [P Nint(1,1) Nint(2,1) Nint(3,1) Nint(4,1)];
NING(P,:) = [P Nintg(1,1) Nintg(2,1) Nintg(3,1) Nintg(4,1)];
% matrice des contrainte
COnt=(Nint/S)*0.001 ; % contrainte
% Matrice des contrainte dens les barres
CONT(P,:) = [P COnt(1,1) COnt(2,1) COnt(3,1) COnt(4,1)];
B=1/L*[-dx/L -dy/L dx/L dy/L];
strain =B*Dpe;
stress=E*strain ;
%vecteur des deformation
def=(Nint/S) /E ; %vecteur des deformation
qui= DEP(2*N1-1,1)*dx/L+DEP(2*N1,1)*dy/L;
qvi=-DEP(2*N1-1,1)*dy/L+DEP(2*N1,1)*dx/L;
quj= DEP(2*N2-1,1)*dx/L+DEP(2*N2,1)*dy/L;
qvj=-DEP(2*N2-1,1)*dy/L+DEP(2*N2,1)*dx/L;
fui=-(qui-quj)*ESL;
fvi=0;
fuj=(-qui+quj)*ESL;

```

```

    fvj=0;
end
MVdep=zeros(NN,3);
for i=1:NN
    MVdep(i,1)=i;
    MVdep(i,2)=(100*DEP(2*i-1));
    MVdep(i,3)=(100*DEP(2*i));
    if abs(MVdep(i,2)) < 0.000000000000001
        MVdep(i,2)=0;
    end
    if abs(MVdep(i,3)) < 0.000000000000001
        MVdep(i,3)=0;
    end
end
MVdep;
% matrice des reactions d'appuis
MRHV=zeros(cpr,3);
j=1;
for i=1:NN
    if CN(i,4)==11
        MRHV(j,1)=i;
        MRHV(j,2)=RHV(2*i-1);
        MRHV(j,3)=RHV(2*i);
        j=j+1;
    end
    if CN(i,4)==10
        MRHV(j,1)=i;
        MRHV(j,2)=RHV(2*i-1);
        MRHV(j,3)=RHV(2*i);
        j=j+1;
    end
    if CN(i,4)==01
        MRHV(j,1)=i;
        MRHV(j,2)=RHV(2*i-1);
        MRHV(j,3)=RHV(2*i);
        j=j+1;
    end
end
% matrice des efforts internes
MNIN=[NIN(:,1) NIN(:,2) -NIN(:,4) CONT(:,2)];
for i=1:NE
    if abs(MNIN(i,2)) < 0.0000000000001
        MNIN(i,2)=0;
    end
    if abs(MNIN(i,3)) < 0.0000000000001
        MNIN(i,3)=0;
    end
    if abs(MNIN(i,4)) < 0.0000000000001
        MNIN(i,4)=0;
    end
end
end

%visualisation des déplacements figure
MDD=zeros(NN,3);
for i=1:NN
    MDD(i,1)=CN(i,1);
    MDD(i,2)=CN(i,2)+(DEP(2*i-1));
    MDD(i,3)=CN(i,3)+(DEP(2*i));
end

%visualisation des déplacements figure
INCN=CN;
hold on;    grid on;    axis equal;
xlabel('AXE DES X');    ylabel('AXE DES Y');
for i=1 :NN
    XE(i)=INCN(i,2);
    YE(i)=INCN(i,3);
end

```

```
axis([XMIN XMAX YMIN YMAX]);
for i=1 :NE
    NUM=EN(i,1);
    X1=INCN(EN(i,2),2);
    Y1=INCN(EN(i,2),3);
    X2=INCN(EN(i,3),2);
    Y2=INCN(EN(i,3),3);
    V1=[X1,X2];
    V2=[Y1,Y2];
    plot(V1,V2,'b-','LineWidth',2);
end
pause(1);
INCN=MDD;
for i=1 :NE
    NUM=EN(i,1);
    X1=INCN(EN(i,2),2);
    Y1=INCN(EN(i,2),3);
    X2=INCN(EN(i,3),2);
    Y2=INCN(EN(i,3),3);
    V1=[X1,X2];
    V2=[Y1,Y2];
    plot(V1,V2,'m-','LineWidth',2);
end
end
```

Apl_calcul_poutres

```

function []=Apl_calcul_poutres()
global NE EN NN CN BF VF DEP KG MG KGDOF vddl MVdep MRHV NMTV VFn VFb VFB
VFVT=zeros(3*NN,1); % matrice matrice des forces total
KG=zeros(NN*3,NN*3)
MG=zeros(NN*3,NN*3)
for P=1 :NE
    E=EN(P,6); % module de young
    A=EN(P,4); % section
    N1=EN(P,2);
    N2=EN(P,3);
    dx=(CN(N2,2)-CN(N1,2));
    dy=(CN(N2,3)-CN(N1,3));
    L=sqrt(dx^2+dy^2); % longueur
    Int=EN(P,5); % inertie
    RO=EN(P,7); %poids volumique
    % CALCUL MATRICE DE RIGIDITE D'UNE POUTRE
    K11=(E*A*(dx/L)^2/L+(12*E*Int*(dy/L)^2/L^3);
    K12=((E*A/L)-(12*E*Int/L^3))*(dx/L)*(dy/L);
    K13=-6*E*Int*(dy/L)/L^2;
    K14=-E*A*(dx/L)^2/L-(12*E*Int*(dy/L)^2/L^3);
    K15=(-E*A/L+(12*E*Int/L^3))*(dx/L)*(dy/L);
    K16=-6*E*Int*(dy/L)/L^2;
    K22=(E*A*(dy/L)^2/L+(12*E*Int*(dx/L)^2/L^3);
    K23=(6*E*Int*(dx/L)/L^2);
    K24=(-E*A/L+12*E*Int/L^3)*(dx/L)*(dy/L);
    K25=-E*A*(dy/L)^2/L-(12*E*Int*(dx/L)^2/L^3);
    K26=(6*E*Int*(dx/L)/L^2);
    K33=4*E*Int/L;
    K34=6*E*Int*(dy/L)/L^2;
    K35=-6*E*Int*(dx/L)/L^2;
    K36=2*E*Int/L;
    K44=(E*A*(dx/L)^2/L+(12*E*Int*(dy/L)^2/L^3);
    K45=(E*A/L-(12*E*Int/L^3))*(dx/L)*(dy/L);
    K46=(6*E*Int*(dy/L)/L^2);
    K55=(E*A*(dy/L)^2/L+(12*E*Int*(dx/L)^2/L^3);
    K56=-6*E*Int*(dx/L)/L^2;
    K66=4*E*Int/L;
    KEi=[K11 K12 K13 K14 K15 K16 ;
        K12 K22 K23 K24 K25 K26 ;
        K13 K23 K33 K34 K35 K36 ;
        K14 K24 K34 K44 K45 K46 ;
        K15 K25 K35 K45 K55 K56 ;
        K16 K26 K36 K46 K56 K66 ];
    Ke=[E*A/L 0 0 -E*A/L 0 0
        0 12*E*Int/L^3 6*E*Int/L^2 0 -12*E*Int/L^3 6*E*Int/L^2
        0 6*E*Int/L^2 4*E*Int/L 0 -6*E*Int/L^2 2*E*Int/L
        -E*A/L 0 0 E*A/L 0 0
        0 -12*E*Int/L^3 -6*E*Int/L^2 0 12*E*Int/L^3 -6*E*Int/L^2
        0 6*E*Int/L^2 2*E*Int/L 0 -6*E*Int/L^2 4*E*Int/L];

    Re=[ dx/L dy/L 0 0 0 0
        -dy/L dx/L 0 0 0 0
        0 0 1 0 0 0
        0 0 0 dx/L dy/L 0
        0 0 0 -dy/L dx/L 0
        0 0 0 0 0 1];

    m=RO*A
    Mle=m*L/420*[140 0 0 70 0 0 ;
        0 156 22*L 0 54 -13*L ;
        0 22*L 4*L^2 0 13*L -3*L^2 ;
        70 0 0 140 0 0 ;
        0 54 13*L 0 156 -22*L ;
        0 -13*L -3*L^2 0 -22*L 4*L^2];

```

%

Me=Re'*Mle*Re;

KE=Re'*Ke*Re;

% ASSEMBLAGE DE LA MATRICE DE RIGIDITE KG

```

KG(3*N1-2,3*N1-2)=KG(3*N1-2,3*N1-2)+KE(1,1);
KG(3*N1-2,3*N1-1)=KG(3*N1-2,3*N1-1)+KE(1,2);
KG(3*N1-2,3*N1) =KG(3*N1-2,3*N1) +KE(1,3);
KG(3*N1-2,3*N2-2)=KG(3*N1-2,3*N2-2)+KE(1,4);
KG(3*N1-2,3*N2-1)=KG(3*N1-2,3*N2-1)+KE(1,5);
KG(3*N1-2,3*N2) =KG(3*N1-2,3*N2) +KE(1,6);
KG(3*N1-1,3*N1-2)=KG(3*N1-1,3*N1-2)+KE(2,1);
KG(3*N1-1,3*N1-1)=KG(3*N1-1,3*N1-1)+KE(2,2);
KG(3*N1-1,3*N1) =KG(3*N1-1,3*N1) +KE(2,3);
KG(3*N1-1,3*N2-2)=KG(3*N1-1,3*N2-2)+KE(2,4);
KG(3*N1-1,3*N2-1)=KG(3*N1-1,3*N2-1)+KE(2,5);
KG(3*N1-1,3*N2) =KG(3*N1-1,3*N2) +KE(2,6);
KG(3*N1,3*N1-2)=KG(3*N1,3*N1-2)+KE(3,1);
KG(3*N1,3*N1-1)=KG(3*N1,3*N1-1)+KE(3,2);
KG(3*N1,3*N1) =KG(3*N1,3*N1) +KE(3,3);
KG(3*N1,3*N2-2)=KG(3*N1,3*N2-2)+KE(3,4);
KG(3*N1,3*N2-1)=KG(3*N1,3*N2-1)+KE(3,5);
KG(3*N1,3*N2) =KG(3*N1,3*N2) +KE(3,6);
KG(3*N2-2,3*N1-2)=KG(3*N2-2,3*N1-2)+KE(4,1);
KG(3*N2-2,3*N1-1)=KG(3*N2-2,3*N1-1)+KE(4,2);
KG(3*N2-2,3*N1) =KG(3*N2-2,3*N1) +KE(4,3);
KG(3*N2-2,3*N2-2)=KG(3*N2-2,3*N2-2)+KE(4,4);
KG(3*N2-2,3*N2-1)=KG(3*N2-2,3*N2-1)+KE(4,5);
KG(3*N2-2,3*N2) =KG(3*N2-2,3*N2) +KE(4,6);
KG(3*N2-1,3*N1-2)=KG(3*N2-1,3*N1-2)+KE(5,1);
KG(3*N2-1,3*N1-1)=KG(3*N2-1,3*N1-1)+KE(5,2);
KG(3*N2-1,3*N1) =KG(3*N2-1,3*N1) +KE(5,3);
KG(3*N2-1,3*N2-2)=KG(3*N2-1,3*N2-2)+KE(5,4);
KG(3*N2-1,3*N2-1)=KG(3*N2-1,3*N2-1)+KE(5,5);
KG(3*N2-1,3*N2) =KG(3*N2-1,3*N2) +KE(5,6);
KG(3*N2,3*N1-2)=KG(3*N2,3*N1-2)+KE(6,1);
KG(3*N2,3*N1-1)=KG(3*N2,3*N1-1)+KE(6,2);
KG(3*N2,3*N1) =KG(3*N2,3*N1) +KE(6,3);
KG(3*N2,3*N2-2)=KG(3*N2,3*N2-2)+KE(6,4);
KG(3*N2,3*N2-1)=KG(3*N2,3*N2-1)+KE(6,5);
KG(3*N2,3*N2) =KG(3*N2,3*N2) +KE(6,6);

```

% ASSEMBLAGE DE LA MATRICE DE MASSE MG Me

```

MG(3*N1-2,3*N1-2)=MG(3*N1-2,3*N1-2)+Me(1,1);
MG(3*N1-2,3*N1-1)=MG(3*N1-2,3*N1-1)+Me(1,2);
MG(3*N1-2,3*N1) =MG(3*N1-2,3*N1) +Me(1,3);
MG(3*N1-2,3*N2-2)=MG(3*N1-2,3*N2-2)+Me(1,4);
MG(3*N1-2,3*N2-1)=MG(3*N1-2,3*N2-1)+Me(1,5);
MG(3*N1-2,3*N2) =MG(3*N1-2,3*N2) +Me(1,6);
MG(3*N1-1,3*N1-2)=MG(3*N1-1,3*N1-2)+Me(2,1);
MG(3*N1-1,3*N1-1)=MG(3*N1-1,3*N1-1)+Me(2,2);
MG(3*N1-1,3*N1) =MG(3*N1-1,3*N1) +Me(2,3);
MG(3*N1-1,3*N2-2)=MG(3*N1-1,3*N2-2)+Me(2,4);
MG(3*N1-1,3*N2-1)=MG(3*N1-1,3*N2-1)+Me(2,5);
MG(3*N1-1,3*N2) =MG(3*N1-1,3*N2) +Me(2,6);
MG(3*N1,3*N1-2)=MG(3*N1,3*N1-2)+Me(3,1);
MG(3*N1,3*N1-1)=MG(3*N1,3*N1-1)+Me(3,2);
MG(3*N1,3*N1) =MG(3*N1,3*N1) +Me(3,3);
MG(3*N1,3*N2-2)=MG(3*N1,3*N2-2)+Me(3,4);
MG(3*N1,3*N2-1)=MG(3*N1,3*N2-1)+Me(3,5);
MG(3*N1,3*N2) =MG(3*N1,3*N2) +Me(3,6);
MG(3*N2-2,3*N1-2)=MG(3*N2-2,3*N1-2)+Me(4,1);
MG(3*N2-2,3*N1-1)=MG(3*N2-2,3*N1-1)+Me(4,2);
MG(3*N2-2,3*N1) =MG(3*N2-2,3*N1) +Me(4,3);
MG(3*N2-2,3*N2-2)=MG(3*N2-2,3*N2-2)+Me(4,4);
MG(3*N2-2,3*N2-1)=MG(3*N2-2,3*N2-1)+Me(4,5);
MG(3*N2-2,3*N2) =MG(3*N2-2,3*N2) +Me(4,6);
MG(3*N2-1,3*N1-2)=MG(3*N2-1,3*N1-2)+Me(5,1);
MG(3*N2-1,3*N1-1)=MG(3*N2-1,3*N1-1)+Me(5,2);
MG(3*N2-1,3*N1) =MG(3*N2-1,3*N1) +Me(5,3);

```

```

MG(3*N2-1,3*N2-2)=MG(3*N2-1,3*N2-2)+Me(5,4);
MG(3*N2-1,3*N2-1)=MG(3*N2-1,3*N2-1)+Me(5,5);
MG(3*N2-1,3*N2) =MG(3*N2-1,3*N2) +Me(5,6);
MG(3*N2,3*N1-2)=MG(3*N2,3*N1-2)+Me(6,1);
MG(3*N2,3*N1-1)=MG(3*N2,3*N1-1)+Me(6,2);
MG(3*N2,3*N1) =MG(3*N2,3*N1) +Me(6,3);
MG(3*N2,3*N2-2)=MG(3*N2,3*N2-2)+Me(6,4);
MG(3*N2,3*N2-1)=MG(3*N2,3*N2-1)+Me(6,5);
MG(3*N2,3*N2) =MG(3*N2,3*N2) +Me(6,6);
syms x
M2N=[1-x/L, 0, 0, x/L
0, 0, 1-((3*(x^2))/(L^2))+((2*(x^3))/(L^3)), x-
((2*(x^2))/L)+((x^3)/(L^2)), 0, (3*(x^2)/(L^2))-((2*(x^3))/(L^3)) -
((x^2)/L)+((x^3)/(L^2))];
Mvolum=M2N'*[0
-RO]
x=[0 L]'
VMvolum=double(subs(Mvolum))
Fvro=[0
-A*RO*L/2
-A*RO*(L^2)/12
0
-A*RO*L/2
A*RO*(L^2)/12];
VFVT(3*N1-2) = VFVT(3*N1-2) +Fvro(1);
VFVT(3*N1-1) = VFVT(3*N1-1) +Fvro(2);
VFVT(3*N1) = VFVT(3*N1) +Fvro(3);
VFVT(3*N2-2) = VFVT(3*N2-2) +Fvro(4);
VFVT(3*N2-1) = VFVT(3*N2-1) +Fvro(5);
VFVT(3*N2) = VFVT(3*N2) +Fvro(6);
end
KGIN=KG
ad=1;
for g=1 :NN
if CN(g,4)==111
cpr=cpr+1 ;
KG(3*CN(g,1)-2,3*CN(g,1)-2)=KG(3*CN(g,1)-2,3*CN(g,1)-2)*1.0e25;
KG(3*CN(g,1)-1,3*CN(g,1)-1)=KG(3*CN(g,1)-1,3*CN(g,1)-1)*1.0e25;
KG(3*CN(g,1),3*CN(g,1))=KG(3*CN(g,1),3*CN(g,1))*1.0e25;
%sprintf('bloker sur X');
vddl(ad)=3*CN(g,1)-2;
vddl(ad+1)=3*CN(g,1)-1;
vddl(ad+2)=3*CN(g,1);
ad=ad+3;
end
if CN(g,4)==010
cpr=cpr+1 ;
KG(3*CN(g,1)-1,3*CN(g,1)-1)=KG(3*CN(g,1)-1,3*CN(g,1)-1)*1.0e25;
%sprintf('bloker sur y');
vddl(ad)=3*CN(g,1)-1;
ad=ad+1;
end
if CN(g,4)==110
cpr=cpr+1 ;
KG(3*CN(g,1)-2,3*CN(g,1)-2)=KG(3*CN(g,1)-2,3*CN(g,1)-2)*1.0e25;
KG(3*CN(g,1)-1,3*CN(g,1)-1)=KG(3*CN(g,1)-1,3*CN(g,1)-1)*1.0e25;
%sprintf('bloker angle');
vddl(ad)=3*CN(g,1)-2;
vddl(ad+1)=3*CN(g,1)-1;
ad=ad+2;
end; end;
KGDOP=KG
DEP=KG\VF
RHV=(KGIN*DEP)-VF; %reaction d'appuis
MVdep=zeros(NN,4); matrice des deplacement en (cm )
for iy=1:NN
MVdep(iy,1)=iy;

```

```

MVdep(iy,2)=(100*DEP(3*iy-2));
MVdep(iy,3)=(100*DEP(3*iy-1));
MVdep(iy,4)=-(DEP(3*iy));
if abs(MVdep(iy,2)) < 0.0000001
    MVdep(iy,2)=0;
end
if abs(MVdep(iy,3)) < 0.0000001
    MVdep(iy,3)=0;
end
if abs(MVdep(iy,4)) < 0.0000001
    MVdep(iy,4)=0;
end ; end;
unit=0.001;
%matrice des reactions d'appuis
MRHV=zeros(cpr,3);
j=1;
for i=1:NN
    if CN(i,4)==111
        MRHV(j,1)=i;
        MRHV(j,2)=unit*RHV(3*i-2);
        MRHV(j,3)=unit*RHV(3*i-1);
        MRHV(j,4)=unit*RHV(3*i);
        j=j+1;
    end
    if CN(i,4)==010
        MRHV(j,1)=i;
        MRHV(j,2)=unit*RHV(3*i-2);
        MRHV(j,3)=unit*RHV(3*i-1);
        MRHV(j,4)=unit*RHV(3*i);
        j=j+1;
    end
    if CN(i,4)==110
        MRHV(j,1)=i;
        MRHV(j,2)=unit*RHV(3*i-2);
        MRHV(j,3)=unit*RHV(3*i-1);
        MRHV(j,4)=unit*RHV(3*i);
        j=j+1;
    end
end
for i=1:size(MRHV(:,1))
    if abs(MRHV(i,2)) < 0.00000001
        MRHV(i,2)=0;
    end
    if abs(MRHV(i,3)) < 0.00000001
        MRHV(i,3)=0;
    end
    if abs(MRHV(i,4)) < 0.00000001
        MRHV(i,4)=0;
    end ; end;
NMTV=zeros(NE*2,5);
for P=1 :NE
    E=EN(P,6);
    A=EN(P,4);
    N1=EN(P,2);
    N2=EN(P,3);
    dx=(CN(N2,2)-CN(N1,2));
    dy=(CN(N2,3)-CN(N1,3));
    L=sqrt(dx^2+dy^2);
    Int=EN(P,5);
    RO=EN(P,7); %poids volumique

Ke=[E*A/L      0      0      -E*A/L      0      0
    0      12*E*Int/L^3  6*E*Int/L^2      0      -12*E*Int/L^3  6*E*Int/L^2
    0      6*E*Int/L^2  4*E*Int/L      0      -6*E*Int/L^2  2*E*Int/L
    -E*A/L      0      0      E*A/L      0      0
    0      -12*E*Int/L^3  -6*E*Int/L^2  0      12*E*Int/L^3  -6*E*Int/L^2
    0      6*E*Int/L^2  2*E*Int/L      0      -6*E*Int/L^2  4*E*Int/L];

```



```

Re= [ dx/L   dy/L   0   0   0   0
      -dy/L  dx/L   0   0   0   0
          0   0   1   0   0   0
          0   0   0   dx/L dy/L  0
          0   0   0  -dy/L dx/L  0
          0   0   0   0   0   1];
sctr=[3*N1-2 3*N1-1 3*N1 3*N2-2 3*N2-1 3*N2];
Dpe=DEP(sctr);
KE=Re'*Ke*Re;
qe=Re*Dpe; % deplacement dans le repere local
if BF~=0
    FNDP = ismember(P,VFB(:,1))
    indexes = find(FNDP)
    VFB(indexes,1)
    if VFB(indexes,1)==1
        Ke*qe
        VFB(sctr)
        effint= Ke*qe-(Re'*VFB(sctr));
        Ni=effint(1,1);
        Ti=effint(2,1);
        Mi=effint(3,1);
        Nj=-effint(4,1);
        Tj=-effint(5,1);
        Mj=effint(6,1);
    else
        effint= Ke*qe;
        Ni=effint(1,1);
        Ti=effint(2,1);
        Mi=effint(3,1);
        Nj=-effint(4,1);
        Tj=-effint(5,1);
        Mj=effint(6,1);
    end
end
if BF==0
    effint= Ke*qe;
    Ni=effint(1,1);
    Ti=effint(2,1);
    Mi=effint(3,1);
    Nj=-effint(4,1);
    Tj=-effint(5,1);
    Mj=effint(6,1);
end
end
NMTV(2*P-1,1)=P;
NMTV(2*P,1)=P;
NMTV(2*P-1,2)=N1;
NMTV(2*P,2)=N2;
NMTV(2*P-1,3)=Ni*unit;
NMTV(2*P,3)=Nj*unit;
NMTV(2*P-1,4)=Ti*unit;
NMTV(2*P,4)=Tj*unit;
NMTV(2*P-1,5)=Mi*unit;
NMTV(2*P,5)=Mj*unit;
end
for jl=1:NE*2
    if abs(NMTV(jl,3)) < 0.000000001
        NMTV(jl,3)=0; end;
    if abs(NMTV(jl,4)) < 0.000000001
        NMTV(jl,4)=0; end;
    if abs(NMTV(jl,5)) < 0.000000001
        NMTV(jl,5)=0; end ; end; end ;
end

```

Apl_Factor_Cholesky

```
function [DD]=Apl_Factor_Cholesky(KG,FG)
    mm=length(KG);
    LKG = zeros(mm,mm);
    for ij=1:mm
        LKG(ij,ij) = sqrt(KG(ij,ij)-(LKG(ij,:) * LKG(ij,:))');
        for k = (ij+1):mm
            LKG(k,ij) = (KG(k,ij) - (LKG(ij,:) * LKG(k,:))')/LKG(ij,ij);
        end
    end
    MY=zeros(3*NNOD,1);
    for ik=1 : NNOD*3
        s=0;
        for in=1:ik-1
            s=s+LKG(ik,in) * MY(in);
        end
        MY(ik)=(FG(ik)-s)/LKG(ik,ik);
    end

    for jk= NNOD*3:-1:1
        s=0;
        for jn=jk+1:NNOD*3
            s=s+LKG(jn,jk)*DEP(jn);
        end
        DEP(jk)=(MY(jk)-s)/LKG(jk,jk);
    end
    DD=DEP'; % vecteur de deplacemet avec choeleski
```

Apl_newmark_implusion

```

function [deplac,temps]=Apl_newmark_implusion(MG,KGDOF,vddl,force,dt,T,amortissement)
Ki =KGDOF; Mi= MG ; [ndk,ndk]=size(Ki); n = length(Mi);
[V,D] = eig(Ki,Mi ); % mode propre
pulsas=sqrt(diag(D)); %Pulsation naturelles rad/s
[pulsas , tri]=sort(pulsas);
switch amortissement
case 'Amortis' %amortissement = 'Amortis' ;
    fprintf(' Amortissement critique selon rayleight \n');
    kxi=0.02;
    alfa=(2*pulsas(1)*pulsas(2)*(pulsas(1)-pulsas(2))*kxi)/(pulsas(1)^2-...
    pulsas(2)^2);
    beta=(2*pulsas(1)*pulsas(2)*(1/pulsas(2)-1/pulsas(1))*kxi)/(pulsas(1)^2-...
    pulsas(2)^2);
    C = alfa*Mi + beta*Ki;
case 'NonAmortis' %amortissement = 'NonAmortis' ;
    C=zeros(ndk,ndk);
end
fddl=[1:ndk]'; fddl(vddl')=[]; % les conditions aux limites
K=Ki(fddl,fddl); M=Mi(fddl,fddl);
delta = 0.5; %parametre de Newmarck
alpha = 0.25*((0.5+delta)^2);
tini = 0.00; % temps initial
tfin = T ;% temps final
nt = fix((tfin-tini)/dt); % longueur de t nombre de pas temps total/dt
% Constantes integration de Newmark
a0 = 1/(alpha*(dt^2)); %avec dt = 0.002 ; % pas de temps dT
a1 = delta/(alpha*dt);
a2 = 1/(alpha*dt);
a3 = (1/(2*alpha))-1;
a4 = (delta/alpha) -1;
a5 = (dt/2)*((delta/alpha)-2);
a6 = dt*(1-delta);
a7 = delta*dt;
%initialisation des matrices deplacement vitesse et acceleration
deplac = zeros(n,nt) ;
vitesse = zeros(n,nt) ;
accler = zeros(n,nt) ;
deplac(:,1) = zeros;
vitesse(:,1) = zeros ;
P=force;
accler(fddl,1) = P(fddl,1) ;
Kchap = K + (a0 * M)+( a1 * C(fddl,fddl));
for t = 1:nt
    Rcap(fddl,t+1)= P(fddl,t+1)+ M*(a0 * deplac(fddl,t) +...
    a2 * vitesse(fddl,t) + a3 * accler(fddl,t))+...
    (C(fddl,fddl)*(a1 * deplac(fddl,t) +...
    a4 * vitesse(fddl,t) + a5 * accler(fddl,t)));
    deplac(fddl,t+1)=Kchap\Rcap(fddl,t+1);
    accler(fddl,t+1)=a0*(deplac(fddl,t+1) -deplac(fddl,t)) -...
    a2*vitesse(fddl,t) - a3 * accler(fddl,t);
    vitesse(fddl,t+1)=vitesse(fddl,t)+a6*accler(fddl,t)+a7*accler(fddl,t+1);
end
figure(1); clf; grid; hold on
temps=[0:dt:tfin]
plot(temps,deplac(4,:), 'r-', 'LineWidth', 2)
xlabel('temps (sec)'), ylabel('displacement (m)');
title('deplacemnt (temps)'); legend('----- : Noeud 2, direction X');
text(0.18,max(deplac(4,:)), 'point Impact', 'Color', 'r');
% -----visualiser les 8 premiers

```

Apl_M_rigidite elementQ8

```

function [] = Apl_M_rigidite ()
global GMXY GME GNE GNN TGNNCL NC TCXY MDD DD KG VFT M1 MNCXY MNCX MNCY MDEP
global VFE VFVT Centr MD MB MASG KGDOF moduleYoung masseVolum coefPoisson
EY=moduleYoung ; %valeur du module de Young
masseVolum ; %valeur de ro masse volumique
vP=coefPoisson ; %valeur coefficient de poisson
E8N=[GME(:,2) GME(:,3) GME(:,4) GME(:,5)...
      GME(:,6) GME(:,7) GME(:,8) GME(:,9)];
M1=[1 0 0 0 ;...
    0 0 0 1 ;...
    0 1 1 0]; % Matrice L
% Matrice D cas Contrainte plane
DCP=(EY/(1-vP^2))*[1 vP 0
                  vP 1 0
                  0 0 (1-vP)/2];
% Matrice D cas Deformation plane
DDP=(EY/((1+vP)*(1-(2*vP))))*[1-vP vP 0
                              vP 1-vP 0
                              0 0 (1-2*vP)/2] ;
msv=[0;-masseVolum];
KG=zeros(2*GNN,2*GNN); % Matrice de Rigidité globale
MASG=zeros(2*GNN,2*GNN); % Matrice de Masse globale
GBB =zeros(3*GNE,16); % matrice globale B pout n=GNE elements
VFVT=zeros(2*GNN,1); % matrice matrice des forces total
MB=zeros(3*4*GNE,16); % matrice des B
MD=zeros(3*GNE,3); % Matrice des D
for I=1:GNE
    nelm = GME(I,1) ;
    Vec_XY_Q8
    r1=-1/sqrt(3);
    s1=-1/sqrt(3);
    [dN,DNrs]= Apl_DNDRS (r1,s1); % dérivées des fonction de formes dN
    [N]=Apl_Fonct_Forme(r1,s1); % Fonction de forme N
    [B,Jdet]= Apl_matrix_B (dN,DNrs,XY);
    B1=B;
    Jdet1=Jdet;
    if GME(I,13)==1
        D = DDP;
    end
    if GME(I,13)<1
        D = DCP;
    end
    KE1 = B1'*D*B1*Jdet1*GME(I,13);
    VFV1=N*msv*GME(I,13)*Jdet1;
    MMAS1=N*N'*masseVolum*GME(I,13)*Jdet1 ; % Matrice de Masse
    r2=-1/sqrt(3);
    s2=+1/sqrt(3);
    [dN,DNrs]= Apl_DNDRS (r2,s2);
    [N]=Apl_Fonct_Forme(r2,s2);
    [B,Jdet]= Apl_matrix_B (dN,DNrs,XY);
    B2=B;
    Jdet2=Jdet;
    if GME(I,13)==1
        %disp(' deformation plane ');
        D = DDP;
    end
    if GME(I,13)<1
        %disp(' Contrainte plane ');
        D = DCP;
    end
    KE2 = B2'*D*B2*Jdet2*GME(I,13) ;
    VFV2=N*msv*GME(I,13)*Jdet2; % charge volumique
    MMAS2=N*N'*masseVolum*GME(I,13)*Jdet2 ; % Matrice de Masse
    r3=+1/sqrt(3);

```

```

s3=+1/sqrt(3);
[dN,DNrs]= Apl_DNDRS (r3,s3);
[N]=Apl_Fonct_Forme(r3,s3);
[B,Jdet]= Apl_matrix_B (dN,DNrs,XY);
B3=B;
Jdet3=Jdet;
if GME(I,13)==1
    D = DDP;
end
if GME(I,13)<1
    D = DCP;
end
KE3 = B3'*D*B3*Jdet3*GME(I,13);
VFV3=N*msv*GME(I,13)*Jdet3; % charge volumique
MMAS3=N*N'*masseVolum*GME(I,13)*Jdet3 ; % Matrice de Masse
r4=+1/sqrt(3);
s4=-1/sqrt(3);
[dN,DNrs]= Apl_DNDRS (r4,s4);
[N]=Apl_Fonct_Forme(r4,s4);
[B,Jdet]= Apl_matrix_B (dN,DNrs,XY);
B4=B;
Jdet4=Jdet;
if GME(I,13)==1
    D = DDP;
end
if GME(I,13)<1
    D = DCP;
end
KE4 = B4'*D*B4*Jdet4*GME(I,13);
VFV4=N*msv*GME(I,13)*Jdet4; % charge volumique
MMAS4=N*N'*masseVolum*GME(I,13)*Jdet4 ; % Matrice de Masse
KE=KE1+KE2+KE3+KE4;
VFV=VFV1+VFV2+VFV3+VFV4;
MMAS=MMAS1+MMAS2+MMAS3+MMAS4 ; % Matrice de Masse global
V=[1 9 2 10 3 11 4 12 5 13 6 14 7 15 8 16];
for i=1:16
    for j=1:16
        KEE(i,j)=KE(V(i),V(j));
        MASE(i,j)=MMAS(V(i),V(j));
    end
end
Mbe=[B1;
      B2;
      B3;
      B4];
% Matrice des B
for t=1:12
    MB(12*(I-1)+t,:)= Mbe(t,:);
end
% Matrice des D
for tt=1:3
    MD(3*(I-1)+tt,:)= D(tt,:);
end
p=E8N(nelm,:); % position matrice adresse des noeuds des element
for ii=1:8
    for jj=1:8
        KG(2*p(ii)-1,2*p(jj)-1)= KG(2*p(ii)-1,2*p(jj)-1)+ KEE(2*ii-1,2*jj-1);
        KG(2*p(ii)-1,2*p(jj)) = KG(2*p(ii)-1,2*p(jj)) + KEE(2*ii-1,2*jj);
        KG(2*p(ii) ,2*p(jj)-1)= KG(2*p(ii) ,2*p(jj)-1)+ KEE(2*ii ,2*jj-1);
        KG(2*p(ii) ,2*p(jj)) = KG(2*p(ii) ,2*p(jj)) + KEE(2*ii ,2*jj) ;
        MASG(2*p(ii)-1,2*p(jj)-1)=MASG(2*p(ii)-1,2*p(jj)-1)+MASE(2*ii-1,2*jj-1);
        MASG(2*p(ii)-1,2*p(jj))=MASG(2*p(ii)-1,2*p(jj))+MASE(2*ii-1,2*jj);
        MASG(2*p(ii),2*p(jj)-1)=MASG(2*p(ii),2*p(jj)-1)+MASE(2*ii,2*jj-1);
        MASG(2*p(ii),2*p(jj))=MASG(2*p(ii),2*p(jj))+MASE(2*ii,2*jj);
    end ; end
% adresse des positions element et noeuds corespondants
for ff=1:8
    VFVT(2*p(ff)-1)=VFV(2*ff-1);

```

```

VFVT(2*p(ff))=VFV(2*ff);
end ; end ;
KGinit=KG; %Matrice de rigidité global sans condition aux limite
%Condition aux limites
for I=1 :GNN
    adr= GMXY(I,1);
    if GMXY(I,4)==11
        KG(2*adr-1,2*adr-1)=KG(2*adr-1,2*adr-1)*1.0e+25;
        KG(2*adr,2*adr)=KG(2*adr,2*adr)*1.0e+25;
        %sprintf('Encastrement')
    end
    if GMXY(I,4)==10
        KG(2*adr-1,2*adr-1)=KG(2*adr-1,2*adr-1)*1.0e+25;
        %sprintf('bloker sur x');
    end
    if GMXY(I,4)==01
        KG(2*adr,2*adr)=KG(2*adr,2*adr)*1.0e+10 ;
        %sprintf('bloker sur y');
    end
end
KGDOF=KG;
VFE= zeros(2*GNN,1);
NC=TGNNCL(:,1); %liste des noeuds chargés
TCXY=[TGNNCL(:,2) TGNNCL(:,3)] ; %matrice valeurs des forces sens x sens y

VFE(2*NC-1)=TCXY(:,1);
VFE(2*NC)= TCXY(:,2);
VFT=VFVT+VFE;
[DD]=Apl_Factor_Cholevsky(KG,FG)
disp('-----');
disp('          Node:          X-Displ          Y-Displ ');
disp('-----');
for i=1:GNN
    depli(i,:)= [i DD(2*i-1) DD(2*i)];
end
disp((depli));
fprintf('          maxX          maxY\n')
fprintf(' %10.7f %10.7f\n',max(depli(:,2)), max(depli(:,3)));
fprintf('          minX          minY\n');
fprintf(' %10.7f %10.7f\n',min(depli(:,2)), min(depli(:,3)));
MDD=zeros(GNN,3);
fact=1;
for i=1:GNN
    valeur_deplacement=[GMXY(i,1) DD(2*i-1) DD(2*i)];
    MDD(i,1)=GMXY(i,1);
    MDD(i,2)=GMXY(i,2)+(fact*DD(2*i-1));
    MDD(i,3)=GMXY(i,3)+(fact*DD(2*i));
end
MDEP=zeros(GNN,5);
for i=1:GNN
    MDEP(i,1)=GMXY(i,1);
    MDEP(i,2)=GMXY(i,2);
    MDEP(i,3)=GMXY(i,3);
    MDEP(i,4)=DD(2*i-1);
    MDEP(i,5)=DD(2*i);
    if abs(MDEP(i,4)) < 0.0000001
        MDEP(i,4)=0.00;
    end
    if abs(MDEP(i,5)) < 0.0000001
        MDEP(i,5)=0.00;
    end
end
Uxydp=fopen('nxydep.out','w');
for i=1:GNN
    fprintf(Uxydp,'%d%s%.3f%s%.3f%s%.10f%s%.10f\n',MDEP(i,1),' ',MDEP(i,2),' ',MDEP(i,3),' ',MDEP(i,4),' ',MDEP(i,5)); end
Uxdp=fopen('Uxdep.out','w');
for i=1:GNN

```

```

    fprintf(Uxdp, '%d%s%.3f%s%.3f%s%.10f\n', MDEP(i,1), ' ', MDEP(i,2), '
', MDEP(i,3), ' ', MDEP(i,4)); end
Uydp=fopen('Uydep.out', 'w'); for i=1:GNN
    fprintf(Uydp, '%d%s%.3f%s%.3f%s%.10f\n', MDEP(i,1), ' ', MDEP(i,2), '
', MDEP(i,3), ' ', MDEP(i,5)); end
%plot figure des deplacement
Apl_plot_depl_node()
%Contrainte deformation aux point de gauss
MNCX=zeros(GNN,4);
MNCY=zeros(GNN,4);
MNCXY=zeros(GNN,4);
fr=fopen('contraintPG.dat', 'w');
fl=fopen('deformationPG.dat', 'w');
iCount =zeros(GNN,1);
for p=1:GNE % nombre d'element
    MMD=MD(3*(p-1)+1:3*(p-1)+3, :);
    E8N(p, :); % position matrice adresse des noeuds des element
    for i=1:8
        VDEC(2*i-1)=DD(2*(E8N(p,i))-1);
        VDEC(2*i)=DD(2*(E8N(p,i)));
    end
    adrb=12*(p-1)+1;
    Vec_XY_Q8
    rc=0; sc=0;
    [dN,DNrs]= Apl_DNDRS (rc,sc);
    [N]=Apl_Fonct_Forme(rc,sc);
    [B,Jdet]= Apl_matrix_B (dN,DNrs,XY);
    Bc=B;
    epsilon=Bc*VDEC';
    Con_sigma=MMD*epsilon;
    StressMidl(p, :)= [p Centr(p,2) Centr(p,2) Con_sigma(1) Con_sigma(2)
Con_sigma(3)];
    EPSL1=MB(adrb:adrb+2, :)*VDEC';
    EPSL2=MB(adrb+3:adrb+5, :)*VDEC';
    EPSL3=MB(adrb+6:adrb+8, :)*VDEC';
    EPSL4=MB(adrb+9:adrb+11, :)*VDEC';
    StrainPG = table(EPSL1, EPSL2, EPSL3, EPSL4);
    VCON1=MMD*EPSL1;
    VCON2=MMD*EPSL2;
    VCON3=MMD*EPSL3;
    VCON4=MMD*EPSL4;
    ContrPG = table(VCON1, VCON2, VCON3, VCON4);
    fprintf(fl, '%10.10f %10.10f %10.10f %10.10f\n', EPSL1, EPSL2, EPSL3, EPSL4) ;
    fprintf(fr, '%10.5f %10.5f %10.5f %10.5f\n', VCON1, VCON2, VCON3, VCON4) ;
    % copie des contraintes 4 points de Gauss dans la matrice gen des contraintes
    sigma=(VCON1+VCON2+VCON3+VCON4)/4;
    Stress(p, :)= [p Centr(p,2) Centr(p,3) sigma(1) sigma(2) sigma(3)];
    MNCX(E8N(p,1),4)=MNCX(E8N(p,1),4)+VCON1(1,1);
    MNCY(E8N(p,1),4)=MNCY(E8N(p,1),4)+VCON1(2,1);
    MNCXY(E8N(p,1),4)=MNCXY(E8N(p,1),4)+VCON1(3,1);
    iCount(E8N(p,1))=iCount(E8N(p,1))+1;
    MNCX(E8N(p,4),4)=MNCX(E8N(p,4),4)+VCON2(1,1);
    MNCY(E8N(p,4),4)=MNCY(E8N(p,4),4)+VCON2(2,1);
    MNCXY(E8N(p,4),4)=MNCXY(E8N(p,4),4)+VCON2(3,1);
    iCount(E8N(p,4))=iCount(E8N(p,4))+1;
    MNCX(E8N(p,3),4)=MNCX(E8N(p,3),4)+VCON3(1,1);
    MNCY(E8N(p,3),4)=MNCY(E8N(p,3),4)+VCON3(2,1);
    MNCXY(E8N(p,3),4)=MNCXY(E8N(p,3),4)+VCON3(3,1);
    iCount(E8N(p,3))=iCount(E8N(p,3))+1;
    MNCX(E8N(p,2),4)=MNCX(E8N(p,2),4)+VCON4(1,1);
    MNCY(E8N(p,2),4)=MNCY(E8N(p,2),4)+VCON4(2,1);
    MNCXY(E8N(p,2),4)=MNCXY(E8N(p,2),4)+VCON4(3,1);
    iCount(E8N(p,2))=iCount(E8N(p,2))+1;
    MNCX(E8N(p,5),4)=MNCX(E8N(p,5),4)+(VCON1(1,1)+VCON4(1,1))/2;
    MNCY(E8N(p,5),4)=MNCY(E8N(p,5),4)+(VCON1(2,1)+VCON4(2,1))/2;
    MNCXY(E8N(p,5),4)=MNCXY(E8N(p,5),4)+(VCON1(3,1)+VCON4(3,1))/2;
    iCount(E8N(p,5))=iCount(E8N(p,5))+1;
    MNCX(E8N(p,6),4)=MNCX(E8N(p,6),4)+(VCON4(1,1)+VCON3(1,1))/2;

```

```

MNCY(E8N(p,6),4)=MNCY(E8N(p,6),4)+(VCON4(2,1)+VCON3(2,1))/2;
MNCXY(E8N(p,6),4)=MNCXY(E8N(p,6),4)+(VCON4(3,1)+VCON3(3,1))/2;
iCount(E8N(p,6))=iCount(E8N(p,6))+1;
MNCX(E8N(p,7),4)=MNCX(E8N(p,7),4)+(VCON3(1,1)+VCON2(1,1))/2;
MNCY(E8N(p,7),4)=MNCY(E8N(p,7),4)+(VCON3(2,1)+VCON2(2,1))/2;
MNCXY(E8N(p,7),4)=MNCXY(E8N(p,7),4)+(VCON3(3,1)+VCON2(3,1))/2;
iCount(E8N(p,7))=iCount(E8N(p,7))+1;
MNCX(E8N(p,8),4)=MNCX(E8N(p,8),4)+(VCON1(1,1)+VCON2(1,1))/2;
MNCY(E8N(p,8),4)=MNCY(E8N(p,8),4)+(VCON1(2,1)+VCON2(2,1))/2;
MNCXY(E8N(p,8),4)=MNCXY(E8N(p,8),4)+(VCON1(3,1)+VCON2(3,1))/2;
iCount(E8N(p,8))=iCount(E8N(p,8))+1;
% ecritures des XY dans la matrice general des contraintes
for k=1 :8
    np=(E8N(p,k));
    MNCX(E8N(p,k),1)=E8N(p,k);
    MNCX(E8N(p,k),2)=GMXY(E8N(p,k),2);
    MNCX(E8N(p,k),3)=GMXY(E8N(p,k),3);
    MNCY(E8N(p,k),1)=E8N(p,k);
    MNCY(E8N(p,k),2)=GMXY(E8N(p,k),2);
    MNCY(E8N(p,k),3)=GMXY(E8N(p,k),3);
    MNCXY(E8N(p,k),1)=E8N(p,k);
    MNCXY(E8N(p,k),2)=GMXY(E8N(p,k),2);
    MNCXY(E8N(p,k),3)=GMXY(E8N(p,k),3);
end end
iCount;
for i=1:GNN
    MNCX(i,4)=MNCX(i,4)/iCount(i);
    MNCY(i,4)=MNCY(i,4)/iCount(i);
    MNCXY(i,4)=MNCXY(i,4)/iCount(i);
end
disp('-----');
disp('          Result Analysis Contraintes  StressMidl          ');
disp('-----');
disp('  Elemt:    1Coord:    2Coord:    X-Strs:    Y-Strs:    XY-Strs  ');
disp('-----');
disp((StressMidl));
disp('-----');
disp('          Result Analysis Contraintes  Stress/4          ');
disp('-----');
disp('  Elemt:    1Coord:    2Coord:    X-Strs:    Y-Strs:    XY-Strs  ');
disp('-----');
disp((Stress));
elmn=(StressMidl(:,1));
Elemt=cellstr(num2str(elmn));
Coord_1=StressMidl(:,2);
Coord_2=StressMidl(:,3);
X_Strs=StressMidl(:,4);
Y_Strs=StressMidl(:,5);
XY_Strs=StressMidl(:,6);
UC=0.001; %unité des contraintes de N/m2 to KN/m2
disp('-----');
disp('          Stress And Strains of Each Elements          ');
disp('-----');
disp('  Noeuds:    X-Strs:    Y-Strs:    XY-Strs:');
for i=1:GNN
    st(i,:)= [i UC*MNCX(i,4) UC*MNCY(i,4) UC*MNCXY(i,4)];end
disp(st);
Noeud=(st(:,1));
Noeuds=cellstr(num2str(Noeud));
X_Strs=st(:,2);
Y_Strs=st(:,3);
XY_Strs=st(:,4);

```


Apl_matrix_B

```
function [B,Jdet]= Apl_matrix_B (dN,DNrs,XY)
global M1
    Jacob = dN*XY;           % la matrice jacobien
    Jinv = inv(Jacob);       % inverse de la matrice Jacobien
    Jdet = det(Jacob) ;      % determinant de la matrice Jacobien
    Jinst=zeros(4,4);
    Jinst(1,1)=Jinv(1,1);
    Jinst(1,2)=Jinv(1,2);
    Jinst(2,1)=Jinv(2,1);
    Jinst(2,2)=Jinv(2,2);
    Jinst(3,3)=Jinv(1,1);
    Jinst(3,4)=Jinv(1,2);
    Jinst(4,3)=Jinv(2,1);
    Jinst(4,4)=Jinv(2,2);
    AJinv = M1*Jinst;
    B = AJinv*DNrs;

End
```

Apl_Fonct_Forme

```
function [N]=Apl_Fonct_Forme(r,s)
    N1=0.25*(-1+r*s+r*r+s*s-r*r*s-r*s*s);
    N2=0.25*(-1-r*s+r*r+s*s-r*r*s+r*s*s);
    N3=0.25*(-1+r*s+r*r+s*s+r*r*s+r*s*s);
    N4=0.25*(-1-r*s+r*r+s*s+r*r*s-r*s*s);
    N5=0.50*(1-s-r*r+r*r*s);
    N6=0.50*(1+r-s*s-r*s*s);
    N7=0.50*(1+s-r*r-s*r*r);
    N8=0.50*(1-r-s*s+r*s*s);
    N=[N1 0
        0 N1
        N2 0
        0 N2
        N3 0
        0 N3
        N4 0
        0 N4
        N5 0
        0 N5
        N6 0
        0 N6
        N7 0
        0 N7
        N8 0
        0 N8];

end
```

Apl_M_rigidite_triglT6

```

function [] = Apl_M_rigidite_trigl()

global GMXY GME GNE GNN TGNNCL NC TCXY DD KG M1 MNCXY MNCX MNCY MDD
global VFE VFVT moduleYoung masseVolum coefPoisson

EY=moduleYoung ; %valeur du module de Young
masseVolum ; %valeur de ro masse volumique
vP=coefPoisson ; %valeur coefficient de poisson
E6N=[GME(:,2) GME(:,3) GME(:,4) GME(:,5)...
      GME(:,6) GME(:,7)]
M1=[1 0 0 0 ;...
    0 0 0 1 ;...
    0 1 1 0]; % Matrice L
% Matrice D cas Contrainte plane
DCP=(EY/(1-vP^2))*[1 vP 0
                  vP 1 0
                  0 0 (1-vP)/2]
% Matrice D cas Deformation plane
DDP=(EY/((1+vP)*(1-(2*vP))))*[1-vP vP 0
                              vP 1-vP 0
                              0 0 (1-2*vP)/2]

%initialisation des matrices genrale
MSVRO=[0;-masseVolum];
KG=zeros(2*GNN,2*GNN); % Matice de Rigidité globale
MASG=zeros(2*GNN,2*GNN); % Matice de Masse globale
NNelem=6 ; % nombre de nodes par element
NPGauss=3 ; % nombre de point de Gauss
GBB =zeros(3*GNE,NNelem*2); % matrice globale B pout n=GNE elements
VFVT=zeros(2*GNN,1); % matrice matrice des forces total
MB=zeros(3*NPGauss*GNE,NNelem*2); % matrice des B
MD=zeros(3*GNE,3); % Matrice des D
% Matrice de rigidité -----
for I=1:GNE
    nelm = GME(I,1) ;
    Vec_XY_T6
    r1=1/6; s1=1/6; w1=1/6;
    [dN,DNrs]= Apl_DN_trigl (r1,s1); % derivées des fonction de formes dN
    [N]=Apl_Fct_Form_trigl(r1,s1); % Fonction de forme N
    [B,Jdet]= Apl_matrix_B (dN,DNrs,XY);
    B1=B;
    Jdet1=Jdet;
    if GME(I,11)==1
        D = DDP; end
    if GME(I,11)<1
        D = DCP; end
    KE1 =w1* B1'*D*B1*Jdet1*GME(I,11)
    VFV1=N*MSVRO*GME(I,11)*Jdet1;
    MMAS1=N*N'*masseVolum*GME(I,11)*Jdet1 ; % Matrice de Masse
    r2=2/3; s2=1/6; w2=1/6;
    [dN,DNrs]= Apl_DN_trigl (r2,s2);
    [N]=Apl_Fct_Form_trigl(r2,s2);
    [B,Jdet]= Apl_matrix_B (dN,DNrs,XY);
    B2=B;
    Jdet2=Jdet;
    if GME(I,11)==1
        D = DDP; % deformation plane
    end
    if GME(I,11)<1
        D = DCP; % contrainte plane
    end
    KE2 =w2* B2'*D*B2*Jdet2*GME(I,11)
    VFV2=N*MSVRO*GME(I,11)*Jdet2; % charge volumique
    MMAS2=N*N'*masseVolum*GME(I,11)*Jdet2 ; % Matrice de Masse
    r3=1/6; s3=2/3; w3=1/6;

```

```

[dN,DNrs]= Apl_DN_trigl (r3,s3);
[N]=Apl_Fct_Form_trigl (r3,s3);
[B,Jdet]= Apl_matrix_B (dN,DNrs,XY);
B3=B;
Jdet3=Jdet;
if GME(I,11)==1
    D = DDP; %disp('Deformation Plane');
end
if GME(I,11)<1
    D = DCP; %disp('Contrainte Plane');
end
KE3 = w3*B3'*D*B3*Jdet3*GME(I,11)
VFV3=N*MSVRO*GME(I,11)*Jdet3; % charge volumique
MMAS3=N*N'*masseVolum*GME(I,11)*Jdet3 ; % Matrice de Masse
KE=KE1+KE2+KE3 ; % Matrice de rigidité global
VFV=VFV1+VFV2+VFV3 ; % Matrice de force volume
MMAS=MMAS1+MMAS2+MMAS3; % Matrice de Masse global
V=[1 7 2 8 3 9 4 10 5 11 6 12 ];
for i=1:12 % 16 a 12
    for j=1:12 %16 a12
        KEE(i,j)=KE(V(i),V(j));
        MASE(i,j)=MMAS(V(i),V(j));
    End ; end
Mbe=[B1;
      B2;
      B3];
% Matrice des B
for t=1:9
    MB(9*(I-1)+t,:)= Mbe(t,:);
end
% Matrice des D
for tt=1:3
    MD(3*(I-1)+tt,:)= D(tt,:);
end
p=E6N(nelm,:); % position matrice adresse des noeuds des elementd
for ii=1:6
    for jj=1:6
        KG(2*p(ii)-1,2*p(jj)-1)=KG(2*p(ii)-1,2*p(jj)-1)+KEE(2*ii-1,2*jj-1);
        KG(2*p(ii)-1,2*p(jj))=KG(2*p(ii)-1,2*p(jj))+KEE(2*ii-1,2*jj);
        KG(2*p(ii),2*p(jj)-1)=KG(2*p(ii),2*p(jj)-1)+KEE(2*ii,2*jj-1);
        KG(2*p(ii),2*p(jj))=KG(2*p(ii),2*p(jj))+KEE(2*ii,2*jj) ;
        MASG(2*p(ii)-1,2*p(jj)-1)=MASG(2*p(ii)-1,2*p(jj)-1)+MASE(2*ii-1,2*jj-1);
        MASG(2*p(ii)-1,2*p(jj))=MASG(2*p(ii)-1,2*p(jj))+MASE(2*ii-1,2*jj);
        MASG(2*p(ii),2*p(jj)-1)=MASG(2*p(ii),2*p(jj)-1)+MASE(2*ii,2*jj-1);
        MASG(2*p(ii),2*p(jj))=MASG(2*p(ii),2*p(jj))+MASE(2*ii,2*jj);
    end ; end
% adresse des positions element et noeuds corespondants
for ff=1:6
    VFVT(2*p(ff)-1)=VFV(2*ff-1);
    VFVT(2*p(ff))=VFV(2*ff);
end ; end

for I=1 :GNN
    adr= GMXY(I,1);
    if GMXY(I,4)==11
        KG(2*adr-1,2*adr-1)=KG(2*adr-1,2*adr-1)*1.0e+25;
        KG(2*adr,2*adr)=KG(2*adr,2*adr)*1.0e+25;
        %sprintf('Encastrement')
    end
    if GMXY(I,4)==10
        KG(2*adr-1,2*adr-1)=KG(2*adr-1,2*adr-1)*1.0e+25;
        %sprintf('bloker sur x');
    end
    if GMXY(I,4)==01
        KG(2*adr,2*adr)=KG(2*adr,2*adr)*1.0e+10 ;
        %sprintf('bloker sur y');
    end
End ; end
VFE= zeros(2*GNN,1);

```

```

NC=TGNNCL(:,1);
TCXY=[TGNNCL(:,2) TGNNCL(:,3)];
VFE(2*NC-1)=TCXY(:,1);
VFE(2*NC)=TCXY(:,2);
VFT=VFVT+VFE;
LTKG=chol(KG);
LKG=LTKG';
disp('Solving Kd=F Equation.....');
MY=zeros(2*GNN,1);
for K=1 : GNN*2
    S=0;
    for I=1:K-1
        S=S+LKG(K,I)*MY(I);
    end
    MY(K)=(VFT(K)-S)/LKG(K,K);
end
for K=GNN*2:-1:1
    S=0;
    for I=K+1:GNN*2
        S=S+LKG(I,K)*DEP(I);
    end
    DEP(K)=(MY(K)-S)/LKG(K,K);
end
DD=DEP' % vecteur de deplacemet
MDEP=zeros(GNN,5);
for i=1:GNN
    MDEP(i,1)=GMXY(i,1);
    MDEP(i,2)=GMXY(i,2);
    MDEP(i,3)=GMXY(i,3);
    MDEP(i,4)=DD(2*i-1);
    MDEP(i,5)=DD(2*i);
    if abs(MDEP(i,4)) < 0.0000001
        MDEP(i,4)=0.00;
    end
    if abs(MDEP(i,5)) < 0.0000001
        MDEP(i,5)=0.00;
    end
end; end
disp('-----');
disp(' Node: 1Coord: 2Coord: X-Displ Y-Displ');
for i=1:GNN
    Plt(i,:)= [MDEP(i,1) MDEP(i,2) MDEP(i,3) MDEP(i,4) MDEP(i,5)];
end
disp(Plt);
figure(1)
Apl_plot_depl_node()
MNCX=zeros(GNN,4);
MNCY=zeros(GNN,4);
MNCXY=zeros(GNN,4);
fr=fopen('contraintPG.dat','w');
for p=1:GNE % nombre d'element
    MMD=MD(3*(p-1)+1:3*(p-1)+3,:);
    E6N(p,:); % position matrice adresse des noeuds des element
    for i=1:6
        VDEC(2*i-1)=DD(2*(E6N(p,i))-1);
        VDEC(2*i)=DD(2*(E6N(p,i))); end
    adrb=9*(p-1)+1;
    EPSL1=MB(adrb:adrb+2,:)*VDEC';
    EPSL2=MB(adrb+3:adrb+5,:)*VDEC';
    EPSL3=MB(adrb+6:adrb+8,:)*VDEC';
    VCON1=MMD*EPSL1;
    VCON2=MMD*EPSL2;
    VCON3=MMD*EPSL3;
    strain_stessGP=table(EPSL1,EPSSL2,EPSSL3,VCON1,VCON2,VCON3);
    ContrPG = table(VCON1,VCON2,VCON3);
    fprintf(fr,'%10.5f %10.5f %10.5f\n',VCON1,VCON2,VCON3);
    %copie des contraintes 4 points de Gauss dans matrice gen contraintes
    MNCX(E6N(p,1),4)=MNCX(E6N(p,1),4)+VCON1(1,1);
    MNCY(E6N(p,1),4)=MNCY(E6N(p,1),4)+VCON1(2,1);

```

```

MNCXY (E6N(p, 1), 4) = MNCXY (E6N(p, 1), 4) + VCON1 (3, 1);
MNCX (E6N(p, 3), 4) = MNCX (E6N(p, 3), 4) + VCON2 (1, 1);
MNCY (E6N(p, 3), 4) = MNCY (E6N(p, 3), 4) + VCON2 (2, 1);
MNCXY (E6N(p, 3), 4) = MNCXY (E6N(p, 3), 4) + VCON2 (3, 1);
MNCX (E6N(p, 5), 4) = MNCX (E6N(p, 5), 4) + VCON3 (1, 1);
MNCY (E6N(p, 5), 4) = MNCY (E6N(p, 5), 4) + VCON3 (2, 1);
MNCXY (E6N(p, 5), 4) = MNCXY (E6N(p, 5), 4) + VCON3 (3, 1);
MNCX (E6N(p, 2), 4) = MNCX (E6N(p, 2), 4) + (VCON1 (1, 1) + VCON2 (1, 1)) / 2;
MNCY (E6N(p, 2), 4) = MNCY (E6N(p, 2), 4) + (VCON1 (2, 1) + VCON2 (2, 1)) / 2;
MNCXY (E6N(p, 2), 4) = MNCXY (E6N(p, 2), 4) + (VCON1 (3, 1) + VCON2 (3, 1)) / 2;
MNCX (E6N(p, 4), 4) = MNCX (E6N(p, 4), 4) + (VCON2 (1, 1) + VCON3 (1, 1)) / 2;
MNCY (E6N(p, 4), 4) = MNCY (E6N(p, 4), 4) + (VCON2 (2, 1) + VCON3 (2, 1)) / 2;
MNCXY (E6N(p, 4), 4) = MNCXY (E6N(p, 4), 4) + (VCON2 (3, 1) + VCON3 (3, 1)) / 2;
MNCX (E6N(p, 6), 4) = MNCX (E6N(p, 6), 4) + (VCON3 (1, 1) + VCON1 (1, 1)) / 2;
MNCY (E6N(p, 6), 4) = MNCY (E6N(p, 6), 4) + (VCON3 (2, 1) + VCON1 (2, 1)) / 2;
MNCXY (E6N(p, 6), 4) = MNCXY (E6N(p, 6), 4) + (VCON3 (3, 1) + VCON1 (3, 1)) / 2;
% ecritures des XY dans la matrice general des contraintes
for k=1 :6
    np=(E6N(p,k));
    MNCX (E6N(p,k), 1) = E6N(p,k);
    MNCX (E6N(p,k), 2) = GMXY (E6N(p,k), 2);
    MNCX (E6N(p,k), 3) = GMXY (E6N(p,k), 3);
    MNCY (E6N(p,k), 1) = E6N(p,k);
    MNCY (E6N(p,k), 2) = GMXY (E6N(p,k), 2);
    MNCY (E6N(p,k), 3) = GMXY (E6N(p,k), 3);
    MNCXY (E6N(p,k), 1) = E6N(p,k);
    MNCXY (E6N(p,k), 2) = GMXY (E6N(p,k), 2);
    MNCXY (E6N(p,k), 3) = GMXY (E6N(p,k), 3);
End; end;
UC= 0.001;
disp('-----');
disp('      Stress And Strains of Each Elements      ');
disp('-----');
disp('      Noeuds:      X-Strs:      Y-Strs:      XY-Strs:');
for i=1:GNN
    st(i,:)=[i UC*MNCX(i,4) UC*MNCY(i,4) UC*MNCXY(i,4)];
end
disp(st);
end

```

Apl_Fct_Form_trigl

```

function [N]=Apl_Fct_Form_trigl(r,s)
    N1=1-3*r-3*s+2*r^2+2*s^2+4*r*s;
    N2=4*r-4*r^2-4*r*s;
    N3=-r+2*r^2;
    N4=4*r*s;
    N5=-s+2*s^2;
    N6=4*s-4*s^2-4*r*s;

    N=[N1 0
        0 N1
        N2 0
        0 N2
        N3 0
        0 N3
        N4 0
        0 N4
        N5 0
        0 N5
        N6 0
        0 N6]; end

```

Apl_Tetra_free

```

global NNOD NELEM NCH MCH az el nip PPs KG FG vddl XYZ EDOF depl MDEP
global ME MXYZ MPRO
Apl_Data_Tetra;
Apl_Graph_Tetra
Eddl=EDOF
KG=zeros(3*NNOD,3*NNOD);
Force_volume_Gen=zeros(3*NNOD,1);
[NELEM,Npe]=size(Eddl);
E8N=Eddl(:,2:Npe);
onp=4;%nombre de points de gauss
nip=onp;
for i=1:NELEM
[XYZ]=Apl_tetre_vecXYZ(MXYZ,ME,i);
nelm = ME(i,1) ;
E = MPRO(ME(i,6),2); % ligne matrice ds propriete
Ro= MPRO(ME(i,6),3); % poids volumique
v = MPRO(ME(i,6),4);
vf=[0 % sens x
0 % sens y
-1]*Ro; % sens z
[gp,w]=Apl_Point_Gauss_tetra(onp);
wp=w';
ksi=gp(:,1);
eta=gp(:,2);
zeta=gp(:,3);
V=XYZ;
Matrice_D
Ke=zeros(4*3,4*3);
fe=zeros(4*3,1);
for kj=1:onp
[J_det,B]=Apl_BJdet_tetra(V,ksi(kj),eta(kj),zeta(kj));
detJ=J_det;
Nn(kj,1)=1-ksi(kj)-eta(kj)-zeta(kj);
Nn(kj,2)=ksi(kj);
Nn(kj,3)=eta(kj);
Nn(kj,4)=zeta(kj);
Nn2(1,1:3:4*3-2)=Nn(kj,:);
Nn2(2,2:3:4*3-1)=Nn(kj,:);
Nn2(3,3:3:4*3) =Nn(kj,:);
Ke=Ke+B'*D*B*detJ*wp(kj);
fe=fe+Nn2'*vf*detJ*wp(kj); % force de volume
end
fnm = sprintf('file_%d.out',i);
fichier = fullfile(pathstr,name,fnm);
dlmwrite(fichier,Ke,'delimiter','\t','precision',10);
p=E8N(nelm,:); % position matrice adresse des noeuds des element
for ii=1:4
for jj=1:4
KG(3*p(ii)-2,3*p(jj)-2) = KG(3*p(ii)-2,3*p(jj)-2) + Ke(3*ii-2,3*jj-2);
KG(3*p(ii)-2,3*p(jj)-1) = KG(3*p(ii)-2,3*p(jj)-1) + Ke(3*ii-2,3*jj-1);
KG(3*p(ii)-2,3*p(jj)) = KG(3*p(ii)-2,3*p(jj)) + Ke(3*ii-2,3*jj);
KG(3*p(ii)-1,3*p(jj)-2) = KG(3*p(ii)-1,3*p(jj)-2) + Ke(3*ii-1,3*jj-2);
KG(3*p(ii)-1,3*p(jj)-1) = KG(3*p(ii)-1,3*p(jj)-1) + Ke(3*ii-1,3*jj-1);
KG(3*p(ii)-1,3*p(jj)) = KG(3*p(ii)-1,3*p(jj)) + Ke(3*ii-1,3*jj);
KG(3*p(ii),3*p(jj)-2) = KG(3*p(ii),3*p(jj)-2) + Ke(3*ii,3*jj-2);
KG(3*p(ii),3*p(jj)-1) = KG(3*p(ii),3*p(jj)-1) + Ke(3*ii,3*jj-1);
KG(3*p(ii),3*p(jj)) = KG(3*p(ii),3*p(jj)) + Ke(3*ii,3*jj);
end
end
for ff=1:4
Force_volume_Gen(3*p(ff)-2)=fe(3*ff-2);
Force_volume_Gen(3*p(ff)-1)=fe(3*ff-1);
Force_volume_Gen(3*p(ff)) =fe(3*ff);

```

```

end
end
VFfn=zeros(3*NNOD,1);
for I=1 :NCH
    NC=MCH(I,1);
    FX=MCH(I,2);
    FY=MCH(I,3);
    FZ=MCH(I,4);
    VFfn(3*NC-2)=VFfn(3*NC-2)+FX;
    VFfn(3*NC-1)=VFfn(3*NC-1)+FY;
    VFfn(3*NC)=VFfn(3*NC)+FZ;
end
FG= VFfn+PPs*Force_volume_Gen      % force globale avec les forces de volume
ad=1;
for g=1 :NNOD
    if MXYZ(g,5)==111
        KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)=KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)*1.0e25;
        KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)=KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)*1.0e25;
        KG(3*MXYZ(g,1),3*MXYZ(g,1))=KG(3*MXYZ(g,1),3*MXYZ(g,1))*1.0e25;
        vddl(ad)=3*MXYZ(g,1)-2;
        vddl(ad+1)=3*MXYZ(g,1)-1;
        vddl(ad+2)=3*MXYZ(g,1);
        ad=ad+3;
    end
    if MXYZ(g,5)==110
        KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)=KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)*1.0e25;
        KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)=KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)*1.0e25;

        vddl(ad)=3*MXYZ(g,1)-2;
        vddl(ad+1)=3*MXYZ(g,1)-1;
        ad=ad+2;
    end
    if MXYZ(g,5)==101
        KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)=KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)*1.0e25;
        KG(3*MXYZ(g,1),3*MXYZ(g,1))=KG(3*MXYZ(g,1),3*MXYZ(g,1))*1.0e25;
        vddl(ad)=3*MXYZ(g,1)-2;
        vddl(ad+1)=3*MXYZ(g,1);
        ad=ad+2;
    end
    if MXYZ(g,5)==011
        KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)=KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)*1.0e25;
        KG(3*MXYZ(g,1),3*MXYZ(g,1))=KG(3*MXYZ(g,1),3*MXYZ(g,1))*1.0e25;
        vddl(ad)=3*MXYZ(g,1)-1;
        vddl(ad+1)=3*MXYZ(g,1);
        ad=ad+2;
    end
    if MXYZ(g,5)==100
        KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)=KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)*1.0e25;
        vddl(ad)=3*MXYZ(g,1)-2;
        ad=ad+1;
    end
    if MXYZ(g,5)==010
        KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)=KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)*1.0e25;
        vddl(ad)=3*MXYZ(g,1)-1;
        ad=ad+1;
    end
    if MXYZ(g,5)==001
        KG(3*MXYZ(g,1),3*MXYZ(g,1))=KG(3*MXYZ(g,1),3*MXYZ(g,1))*1.0e25;
        vddl(ad)=3*MXYZ(g,1);
        ad=ad+1;
    end
end
end
DvD = KG\FG;
Depl= DvD
for i = 1:NELEM
    ed(i,1:(Npe-1))=depl(E8N(i,:))';
end
MDD=zeros(NNOD,4);

```

```

fact=1; % facteur d'amplification
for i=1:NNOD
    MDD(i,1)=MXYZ(i,1);
    MDD(i,2)=MXYZ(i,2)+(fact*depl(3*i-2));
    MDD(i,3)=MXYZ(i,3)+(fact*depl(3*i-1));
    MDD(i,4)=MXYZ(i,4)+(fact*depl(3*i));
end
for p = 1:NELEM
    E=MPRO(ME(p,6),2);
    Ro=MPRO(ME(p,6),3);
    v=MPRO(ME(p,6),4);
    E8N(p,:); % position matrice adresse des noeuds des element
    for ig=1:4
        VDEC(3*ig-2)=depl(3*(E8N(p,ig))-2);
        VDEC(3*ig-1)=depl(3*(E8N(p,ig))-1);
        VDEC(3*ig)=depl(3*(E8N(p,ig)));
    end
    [gp,w]=Apl_Point_Gauss_tetra(omp);
    wp=w';
    ksi=gp(:,1);
    eta=gp(:,2);
    zeta=gp(:,3);
    [XYZ]=Apl_tetre_vecXYZ(MXYZ,ME,p);
    Stress=[]; Epsilone=[];
    Matrice_D
    for i=1:omp
        [Nn,dNr]=Apl_dNr_tetra(omp,ksi(i),eta(i),zeta(i));
        blablacula=[ksi(i) eta(i) zeta(i)];
        [J_det,B]=Apl_BJdet_tetra(XYZ,ksi(i),eta(i),zeta(i));
        epsy=B*VDEC(1,:);
        Epsilone=[Epsilone; epsy'];
        Stress=[Stress; (D*epsy)'];
    end
    disp(Stress)
end

for p = 1:NELEM
    E8N(p,:); % position matrice adresse des noeuds des element
    for i=1:4
        VDEC(3*i-2)=depl(3*(E8N(p,i))-2);
        VDEC(3*i-1)=depl(3*(E8N(p,i))-1);
        VDEC(3*i)=depl(3*(E8N(p,i)));
    end
    irc=1;
    [gp,w]=Apl_Point_Gauss_tetra(irc);
    wp=w';
    ksi=gp(:,1);
    eta=gp(:,2);
    zeta=gp(:,3);
    [Nn,dNr]=Apl_dNr_tetra(irc,ksi,eta,zeta);
    [XYZ]=Apl_tetre_vecXYZ(MXYZ,ME,p);
    Stressc=[]; Epsilone=[];
    E =MPRO(ME(p,6),2);
    Ro =MPRO(ME(p,6),3);
    v =MPRO(ME(p,6),4);
    Matrice_D

    for i=1:irc
        [J_det,B]=Apl_BJdet_tetra(XYZ,ksi(i),eta(i),zeta(i));
        epsy=B*VDEC(1,:);
        Epsilone=[Epsilone; epsy'];
        Stressc=[Stressc; (D*epsy)'];
    end
    Mat_CPes(p,:)=Stressc;
end
disp(Mat_CPes)

```


Apl_matrice_D

```

function [D]=Apl_matrice_D(type,E,v)

switch type
case 'trid_3d' % type = ' trid_3d ' ;

D=E/(1+v)/(1-2*v)*[1-v  v  v  0  0  0
                  v  1-v  v  0  0  0
                  v  v  1-v  0  0  0
                  0  0  0  (1-2*v)/2  0  0
                  0  0  0  0  (1-2*v)/2  0
                  0  0  0  0  0  (1-2*v)/2];

case 'contr_plane'
DP=(E/(1-v^2))*[1  v  0
                v  1  0
                0  0  (1-v)/2]

case 'deform_plane'

D=(E/((1+v)*(1-(2*v))))*[1-v  v  0
                        v  1-v  0
                        0  0  (1-2*v)/2]

end

```

Apl_Point_Gauss_tetra

```
function [s,wt]=Apl_Point_Gauss_tetra(ir)
if ir==1
    s(1,1)=0.25;
    s(1,2)=0.25;
    s(1,3)=0.25;
    wt(1)=1.0./6.0;
elseif ir==4
    s(1,1)=0.58541020;
    s(1,2)=0.13819660;
    s(1,3)=s(1,2);
    s(2,2)=s(1,1);
    s(2,3)=s(1,2);
    s(2,1)=s(1,2);
    s(3,3)=s(1,1);
    s(3,1)=s(1,2);
    s(3,2)=s(1,2);
    s(4,1)=s(1,2);
    s(4,2)=s(1,2);
    s(4,3)=s(1,2);
    wt([1:4])=0.25./6.0;
elseif ir==5
    s(1,1)=0.25;
    s(1,2)=0.25;
    s(1,3)=0.25;
    s(2,1)=0.5;
    s(2,2)=1.0./6.0;
    s(2,3)=s(2,2);
    s(3,2)=0.5;
    s(3,3)=1.0./6.0;
    s(3,1)=s(3,3);
    s(4,3)=0.5;
    s(4,1)=1.0./6.0;
    s(4,2)=s(4,1);
    s(5,1)=1.0./6.0;
    s(5,2)=s(5,1);
    s(5,3)=s(5,1);
    wt(1)=-0.8;
    wt(2)=9.0./20.0;
    wt([3:5])=wt(2);
    wt=wt./6.0;
else
    disp('erreur valeur de ir pour element tetraedre')
    return
end ; end
```

Apl_BJdet_tetra

```
function [J_det, B]=Apl_BJdet_tetra(V,r,s,t)

x=V(:,1); y=V(:,2); z=V(:,3);
dNr=[-1; 1; 0; 0];
dNs=[-1; 0; 1; 0];
dNt=[-1; 0; 0; 1];

J=[x'*dNr y'*dNr z'*dNr
   x'*dNs y'*dNs z'*dNs
   x'*dNt y'*dNt z'*dNt];
J_det=det(J);

dNx=zeros(4,1); dNy=zeros(4,1); dNz=zeros(4,1);
for i=1:1:4
    NN=J\[dNr(i) ; dNs(i) ; dNt(i)];
    dNx(i)=NN(1);
    dNy(i)=NN(2);
    dNz(i)=NN(3);
end

B=[ dNx(1) 0      0      dNx(2) 0      0      dNx(3) 0      0      dNx(4) 0      0
    0      dNy(1) 0      0      dNy(2) 0      0      dNy(3) 0      0      dNy(4) 0
    0      0      dNz(1) 0      0      dNz(2) 0      0      dNz(3) 0      0      dNz(4)
    dNy(1) dNx(1) 0      dNy(2) dNx(2) 0      dNy(3) dNx(3) 0      dNy(4) dNx(4) 0
    0      dNz(1) dNy(1) 0      dNz(2) dNy(2) 0      dNz(3) dNy(3) 0      dNz(4) dNy(4)
    dNz(1) 0      dNx(1) dNz(2) 0      dNx(2) dNz(3) 0      dNx(3) dNz(4) 0      dNx(4) ];
end
```

Cprinc1

```
function [Sig_PRINC] =Cprinc1(sigma)

% calcul des Contraintes principales
% sigma1,sigma2,sigma3 a partir du tenseur de contraintes
% sigma=[sigx sigy sigz txy tyz tzx]

sigx=sigma(1);
sigy=sigma(2);
sigz=sigma(3);
txy=sigma(4);tyx=txy;
tyz=sigma(5);tzy=tyz;
txz=sigma(6);txz=txz;
I1=sigx+sigy+sigz;
I2=sigx*sigy+sigy*sigz+sigz*sigx-txy^2-tyz^2-txz^2;
I3=sigx*sigy*sigz+2*txy*tyz*txz-(sigx*tyz^2+sigy*txz^2+sigz*txy^2);

p=[1 -I1 I2 -I3];
r=sort (roots(p), 'descend');
sig1=r(1); sig2=r(2); sig3=r(3);
Sig_PRINC=[sig1 sig2 sig3 ];
end
```

Apl_cub_8N

```

function []=Apl_cub_8N(ir)

global NNOD NELEM ME MXYZ MPRO EDOF KG FG depl NCH MCH ir PPs

Apl_Data_Cub_8N;
Apl_Graph_Cub_8N;
% ir =2;
Eddl=EDOF;
KG=zeros(3*NNOD,3*NNOD);
FvG=zeros(3*NNOD,1);
[NELEM,Npe]=size(Eddl);
E8N=Eddl(:,2:Npe);
for i=1:NELEM
[XY]=Apl_cub8_vecXY(MXYZ,ME,i);
nelm = ME(i,1);
E=MPRO(ME(i,10),2);
Ro=MPRO(ME(i,10),3);
v=MPRO(ME(i,10),4);
vf=[0 % sens x
0 % sens y
-1]*Ro; % sens z
[vdN]=Apl_Fonct_Forme_cub8()
ngp=ir*ir*ir; %nombre de ponits de gauss
[gp,w]=Apl_cub8_Points_Gauss(ir);
wp=w(:,1).*w(:,2).*w(:,3);
ksi=gp(:,1); eta=gp(:,2); zeta=gp(:,3); nbpg=ngp*3;
Ke=zeros(24,24);
fe=zeros(24,1);
Jacb = dNr*XY;
Matrice_D
for ip=1:ngp
indx=[ 3*ip-2; 3*ip-1; 3*ip ];
detJ=det(Jacb(indx,:));
if detJ<10*eps
disp('Determinant du Jacobi egale a zero!');
end
JTinv=inv(Jacb(indx,:));
dNx=JTinv*dNr(indx,:);
B(1,1:3:24-2)=dNx(1,:);
B(2,2:3:24-1)=dNx(2,:);
B(3,3:3:24) =dNx(3,:);
B(4,1:3:24-2)=dNx(2,:);
B(4,2:3:24-1)=dNx(1,:);
B(5,1:3:24-2)=dNx(3,:);
B(5,3:3:24) =dNx(1,:);
B(6,2:3:24-1)=dNx(3,:);
B(6,3:3:24) =dNx(2,:);
Nn2(1,1:3:24-2)=Nn(ip,:);
Nn2(2,2:3:24-1)=Nn(ip,:);
Nn2(3,3:3:24) =Nn(ip,:);
Ke=Ke+B'*D*B*detJ*wp(ip);
fe=fe+Nn2'*vf*detJ*wp(ip);
end
p=E8N(nelm,:);
for ii=1:8
for jj=1:8
KG(3*p(ii)-2,3*p(jj)-2)=KG(3*p(ii)-2,3*p(jj)-2)+Ke(3*ii-2,3*jj-2);
KG(3*p(ii)-2,3*p(jj)-1)=KG(3*p(ii)-2,3*p(jj)-1)+Ke(3*ii-2,3*jj-1);
KG(3*p(ii)-2,3*p(jj)) =KG(3*p(ii)-2,3*p(jj)) +Ke(3*ii-2,3*jj);
KG(3*p(ii)-1,3*p(jj)-2)=KG(3*p(ii)-1,3*p(jj)-2)+Ke(3*ii-1,3*jj-2);
KG(3*p(ii)-1,3*p(jj)-1)=KG(3*p(ii)-1,3*p(jj)-1)+Ke(3*ii-1,3*jj-1);
KG(3*p(ii)-1,3*p(jj)) =KG(3*p(ii)-1,3*p(jj)) +Ke(3*ii-1,3*jj);
KG(3*p(ii),3*p(jj)-2) =KG(3*p(ii),3*p(jj)-2)+Ke(3*ii,3*jj-2);
KG(3*p(ii),3*p(jj)-1) =KG(3*p(ii),3*p(jj)-1)+Ke(3*ii,3*jj-1);
KG(3*p(ii),3*p(jj)) =KG(3*p(ii),3*p(jj)) +Ke(3*ii,3*jj);
end
end

```

```

end
for ff=1:8
    FvG(3*p(ff)-2)=fe(3*ff-2);
    FvG(3*p(ff)-1)=fe(3*ff-1);
    FvG(3*p(ff))=fe(3*ff);
end; end
VFn=zeros(3*NNOD,1);
for I=1 :NCH
    NC=MCH(I,1);
    FX=MCH(I,2);
    FY=MCH(I,3);
    FZ=MCH(I,4);
    VFn(3*NC-2)=VFn(3*NC-2)+FX;
    VFn(3*NC-1)=VFn(3*NC-1)+FY;
    VFn(3*NC)=VFn(3*NC)+FZ;
end
FG= VFn+PPs*FvG; % force globale
% Condition aux limites
ad=1;
for g=1 :NNOD
    if MXYZ(g,5)==111
        KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)=KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)*1.0e25;
        KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)=KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)*1.0e25;
        KG(3*MXYZ(g,1),3*MXYZ(g,1))=KG(3*MXYZ(g,1),3*MXYZ(g,1))*1.0e25;
        vddl(ad)=3*MXYZ(g,1)-2;
        vddl(ad+1)=3*MXYZ(g,1)-1;
        vddl(ad+2)=3*MXYZ(g,1);
        ad=ad+3;
    end
    if MXYZ(g,5)==110
        KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)=KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)*1.0e25;
        KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)=KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)*1.0e25;

        vddl(ad)=3*MXYZ(g,1)-2;
        vddl(ad+1)=3*MXYZ(g,1)-1;
        ad=ad+2;
    end
    if MXYZ(g,5)==101
        KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)=KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)*1.0e25;
        KG(3*MXYZ(g,1),3*MXYZ(g,1))=KG(3*MXYZ(g,1),3*MXYZ(g,1))*1.0e25;

        vddl(ad)=3*MXYZ(g,1)-2;
        vddl(ad+1)=3*MXYZ(g,1);
        ad=ad+2;
    end
    if MXYZ(g,5)==011
        KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)=KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)*1.0e25;
        KG(3*MXYZ(g,1),3*MXYZ(g,1))=KG(3*MXYZ(g,1),3*MXYZ(g,1))*1.0e25;

        vddl(ad)=3*MXYZ(g,1)-1;
        vddl(ad+1)=3*MXYZ(g,1);
        ad=ad+2;
    end
    if MXYZ(g,5)==100
        KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)=KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-
2)*1.0e25;
        vddl(ad)=3*MXYZ(g,1)-2;
        ad=ad+1;
    end
    if MXYZ(g,5)==010
        KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)=KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-
1)*1.0e25;
        vddl(ad)=3*MXYZ(g,1)-1;
        ad=ad+1;
    end
    if MXYZ(g,5)==001
        KG(3*MXYZ(g,1),3*MXYZ(g,1))=KG(3*MXYZ(g,1),3*MXYZ(g,1))*1.0e25;
        vddl(ad)=3*MXYZ(g,1);

```

```

        ad=ad+1;
    end; end
MDD=zeros(NNOD,4);
fact=1;
for i=1:NNOD
    MDD(i,1)=MXYZ(i,1);
    MDD(i,2)=MXYZ(i,2)+(fact*depl(3*i-2));
    MDD(i,3)=MXYZ(i,3)+(fact*depl(3*i-1));
    MDD(i,4)=MXYZ(i,4)+(fact*depl(3*i));
end
for p = 1:NELEM
    E8N(p,:);
    for i=1:8
        VDEC(3*i-2)=depl(3*(E8N(p,i))-2);
        VDEC(3*i-1)=depl(3*(E8N(p,i))-1);
        VDEC(3*i)=depl(3*(E8N(p,i)));
    end
    trans_vdec=VDEC';
    ngp=ir*ir*ir; nbpg=ngp*3;
    gp,w]=Apl_cub8_Points_Gauss(ir);
    wp=w(:,1).*w(:,2).*w(:,3);
    ksi=gp(:,1);
    eta=gp(:,2);
    zeta=gp(:,3);
    [Nn,dNr]=fonction_forme(np,ksi,eta,zera)
    ed=VDEC;
    vect_vedt=VDEC;
    [XY]=Apl_cub8_vecXY(MXYZ,ME,p);
    ex=XY(:,1)';
    ey=XY(:,2)';
    ez=XY(:,3)';
    indis=1; es=[]; et=[]; ie=1;
    E=MPRO(ME(p,10),2);
    v=MPRO(ME(p,10),4);
    for ied=1:size(ed,1)
        JT=dNr*[ex(1,:);ey(1,:);ez(1,:)]';
        Matrice_D
        for i=1:ngp
            indx=[ 3*i-2; 3*i-1; 3*i ];
            detJ=det(JT(indx,:));
            if detJ<10*eps
                disp('le determinant du Jacobien presque egale a zero!')
            end
            JTinv=inv(JT(indx,:));
            dNx=JTinv*dNr(indx,:);
            B(1,1:3:24-2)=dNx(1,:);
            B(2,2:3:24-1)=dNx(2,:);
            B(3,3:3:24) =dNx(3,:);
            B(4,1:3:24-2)=dNx(2,:);
            B(4,2:3:24-1)=dNx(1,:);
            B(5,1:3:24-2)=dNx(3,:);
            B(5,3:3:24) =dNx(1,:);
            B(6,2:3:24-1)=dNx(3,:);
            B(6,3:3:24) =dNx(2,:);
            ee=B*ed(ied,:); et=[et; ee']; es=[es; (D*ee)'];
        end; ie=ie+ indis; end
        disp(es);
    for p = 1:NELEM
        E8N(p,:); % position matrice adresse des noeuds des element
        for i=1:8
            VDEC(3*i-2)=depl(3*(E8N(p,i))-2);
            VDEC(3*i-1)=depl(3*(E8N(p,i))-1);
            VDEC(3*i)=depl(3*(E8N(p,i)));
        end
        trans_vdec=VDEC';
        irc=1;
        ngp=irc*irc*irc;
        [gp,w]=Apl_cub8_Points_Gauss(irc);

```

```

wp=w(:,1).*w(:,2);
ksi=gp(:,1); eta=gp(:,1); zeta=gp(:,1);
nbpg=ngp*3;
np=1;
[Nn,dNr]=fonction_forme(np,ksi,eta,zeta)
ed=VDEC;

[XY]=Apl_cub8_vecXY(MXYZ,ME,p);
ex=XY(:,1)';
ey=XY(:,2)';
ez=XY(:,3)';
incie=1;
es=[]; et=[];
ie=1;
E=MPRO(ME(p,10),2);
v=MPRO(ME(p,10),4);
for ied=1:size(ed,1);
JT=dNr*[ex(1,:);ey(1,:);ez(1,:)]';
Matrice_D
for i=1:ngp
    indx=[ 3*i-2; 3*i-1; 3*i ];
    detJ=det(JT(indx,:));
    if detJ<10*eps
        disp('le determinant du Jacobien presque egale a zero!')
    end
    JTinv=inv(JT(indx,:));
    dNx=JTinv*dNr(indx,:);
    B(1,1:3:24-2)=dNx(1,:);
    B(2,2:3:24-1)=dNx(2,:);
    B(3,3:3:24) =dNx(3,:);
    B(4,1:3:24-2)=dNx(2,:);
    B(4,2:3:24-1)=dNx(1,:);
    B(5,1:3:24-2)=dNx(3,:);
    B(5,3:3:24) =dNx(1,:);
    B(6,2:3:24-1)=dNx(3,:);
    B(6,3:3:24) =dNx(2,:);
    ee=B*ed(ied,:)' ; et=[et; ee']; es=[es; (D*ee)'];
end; ie=ie+incie; end;
Mat_CPes(p,:)=es;
bpg=Nn*XY;
coodC=[bpg(1,1) bpg(1,2) bpg(1,3)];
cood_centre(p,:)=coodC;
end
File_result=[cood_centre Mat_CPes]
end

```

Apl_Fonct_Forme_cub8

```

function [vdN]=Apl_Fonct_Forme_cub8()

syms r s t;
% fonction de forme Utilise
N1=(1-r).*(1-s).*(1-t)/8;
N2=(1+r).*(1-s).*(1-t)/8;
N3=(1+r).*(1+s).*(1-t)/8;
N4=(1-r).*(1+s).*(1-t)/8;
N5=(1-r).*(1-s).*(1+t)/8;
N6=(1+r).*(1-s).*(1+t)/8;
N7=(1+r).*(1+s).*(1+t)/8;
N8=(1-r).*(1+s).*(1+t)/8;
vdndr1 = diff(N1)./diff(r);
vdndr2 = diff(N2)./diff(r);
vdndr3 = diff(N3)./diff(r);
vdndr4 = diff(N4)./diff(r);
vdndr5 = diff(N5)./diff(r);
vdndr6 = diff(N6)./diff(r);
vdndr7 = diff(N7)./diff(r);
vdndr8 = diff(N8)./diff(r);
vdnds1 = diff(N1)./diff(s);
vdnds2 = diff(N2)./diff(s);
vdnds3 = diff(N3)./diff(s);
vdnds4 = diff(N4)./diff(s);
vdnds5 = diff(N5)./diff(s);
vdnds6 = diff(N6)./diff(s);
vdnds7 = diff(N7)./diff(s);
vdnds8 = diff(N8)./diff(s);
vdndt1 = diff(N1)./diff(t);
vdndt2 = diff(N2)./diff(t);
vdndt3 = diff(N3)./diff(t);
vdndt4 = diff(N4)./diff(t);
vdndt5 = diff(N5)./diff(t);
vdndt6 = diff(N6)./diff(t);
vdndt7 = diff(N7)./diff(t);
vdndt8 = diff(N8)./diff(t);
vdN=[vdndr1 vdndr2 vdndr3 vdndr4 vdndr5 vdndr6 vdndr7 vdndr8;...
vdnds1 vdnds2 vdnds3 vdnds4 vdnds5 vdnds6 vdnds7 vdnds8
vdndt1 vdndt2 vdndt3 vdndt4 vdndt5 vdndt6 vdndt7 vdndt8] ;
end

```


Apl_Cube_20N_free

```

function []=Apl_Cube_20N_free()
Eddl=EDOF;
KG=zeros(3*NNOD,3*NNOD);
Force_volume_Gen=zeros(3*NNOD,1);
[NELEM,Npe]=size(Eddl);
E8N=Eddl(:,2:Npe);
for i=1:NELEM
[XY]=Apl_VecXYZ(MXYZ,ME,i);
nelm = ME(i,1) ;
E = MPRO(ME(i,22),2);
Ro= MPRO(ME(i,22),3);
v = MPRO(ME(i,22),4);
vf=[0
    0
    -1]*Ro;
ngp=ir*ir*ir; %nombre de points de gauss
[gp,w]=Apl_Points_Gauss(ir);
wp=w(:,1).*w(:,2).*w(:,3);
ksi=gp(:,1);
eta=gp(:,2);
zeta=gp(:,3);
nbp=ngp*3;
[Nn,dNr]=Apl_Fonct_Forme(nbp,ksi,eta,zeta);
Ke=zeros(60,60);
fe=zeros(60,1);
Jacb = dNr*XY; % la matrice jacobien
Matrice_D
for f=1:ngp
    indx=[ 3*f-2; 3*f-1; 3*f ];
    detJ=det(Jacb(indx,:));
    % mouchards
    if detJ<10*eps
        disp('Determinant du Jacobi egale a zero!');
    end
    JTinv=inv(Jacb(indx,:));
    dNx=JTinv*dNr(indx,:);
    B(1,1:3:60-2)=dNx(1,:);
    B(2,2:3:60-1)=dNx(2,:);
    B(3,3:3:60) =dNx(3,:);
    B(4,1:3:60-2)=dNx(2,:);
    B(4,2:3:60-1)=dNx(1,:);
    B(5,1:3:60-2)=dNx(3,:);
    B(5,3:3:60) =dNx(1,:);
    B(6,2:3:60-1)=dNx(3,:);
    B(6,3:3:60) =dNx(2,:);
    Nn2(1,1:3:60-2)=Nn(f,:);
    Nn2(2,2:3:60-1)=Nn(f,:);
    Nn2(3,3:3:60) =Nn(f,:);
    Ke=Ke+B'*D*B*detJ*wp(f);
    fe=fe+Nn2'*vf*detJ*wp(f); % force de volume
end
p=E8N(nelm,:); % position matrice adresse des noeuds des element
for ii=1:20
    for jj=1:20
        KG(3*p(ii)-2,3*p(jj)-2)= KG(3*p(ii)-2,3*p(jj)-2)+ Ke(3*ii-2,3*jj-2);
        KG(3*p(ii)-2,3*p(jj)-1)= KG(3*p(ii)-2,3*p(jj)-1)+ Ke(3*ii-2,3*jj-1);
        KG(3*p(ii)-2,3*p(jj)) = KG(3*p(ii)-2,3*p(jj)) + Ke(3*ii-2,3*jj);
        KG(3*p(ii)-1,3*p(jj)-2)= KG(3*p(ii)-1,3*p(jj)-2)+ Ke(3*ii-1,3*jj-2);
        KG(3*p(ii)-1,3*p(jj)-1)= KG(3*p(ii)-1,3*p(jj)-1)+ Ke(3*ii-1,3*jj-1);
        KG(3*p(ii)-1,3*p(jj)) = KG(3*p(ii)-1,3*p(jj)) + Ke(3*ii-1,3*jj);
        KG(3*p(ii),3*p(jj)-2) = KG(3*p(ii),3*p(jj)-2)+ Ke(3*ii,3*jj-2);
        KG(3*p(ii),3*p(jj)-1) = KG(3*p(ii),3*p(jj)-1)+ Ke(3*ii,3*jj-1);
        KG(3*p(ii),3*p(jj)) = KG(3*p(ii),3*p(jj)) + Ke(3*ii,3*jj) ;
    end; end
for ff=1:20

```

```

Force_volume_Gen(3*p(ff)-2)=fe(3*ff-2);
Force_volume_Gen(3*p(ff)-1)=fe(3*ff-1);
Force_volume_Gen(3*p(ff)) =fe(3*ff);
End; end
VFfn=zeros(3*NNOD,1);
for I=1 :NCH
    NC=MCH(I,1);
    FX=MCH(I,2);
    FY=MCH(I,3);
    FZ=MCH(I,4);
    VFfn(3*NC-2)=VFfn(3*NC-2)+FX;
    VFfn(3*NC-1)=VFfn(3*NC-1)+FY;
    VFfn(3*NC)=VFfn(3*NC)+FZ;
end
FG= VFfn+PPs*Force_volume_Gen; % force globale avec les forces de volume
LKIU=[VFfn Force_volume_Gen FG]
ad=1;
for g=1 :NNOD
    if MXYZ(g,5)==111
        KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)=KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)*1.0e25;
        KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)=KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)*1.0e25;
        KG(3*MXYZ(g,1),3*MXYZ(g,1)) =KG(3*MXYZ(g,1),3 *MXYZ(g,1)) *1.0e25;
        vddl(ad) =3*MXYZ(g,1)-2;
        vddl(ad+1)=3*MXYZ(g,1)-1;
        vddl(ad+2)=3*MXYZ(g,1);
        ad=ad+3;
    end
    if MXYZ(g,5)==110
        KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)=KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)*1.0e25;
        KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)=KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)*1.0e25;
        vddl(ad) =3*MXYZ(g,1)-2;
        vddl(ad+1)=3*MXYZ(g,1)-1;
        ad=ad+2;
    end
    if MXYZ(g,5)==101
        KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)=KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)*1.0e25;
        KG(3*MXYZ(g,1),3*MXYZ(g,1)) =KG(3*MXYZ(g,1),3*MXYZ(g,1)) *1.0e25;
        vddl(ad) =3*MXYZ(g,1)-2;
        vddl(ad+1)=3*MXYZ(g,1);
        ad=ad+2;
    end
    if MXYZ(g,5)==011
        KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)=KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)*1.0e25;
        KG(3*MXYZ(g,1),3*MXYZ(g,1)) =KG(3*MXYZ(g,1),3*MXYZ(g,1)) *1.0e25;
        vddl(ad) =3*MXYZ(g,1)-1;
        vddl(ad+1)=3*MXYZ(g,1);
        ad=ad+2;
    end
    if MXYZ(g,5)==100
        KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)=KG(3*MXYZ(g,1)-2,3*MXYZ(g,1)-2)*1.0e25;
        vddl(ad)=3*MXYZ(g,1)-2;
        ad=ad+1;
    end
    if MXYZ(g,5)==010
        KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)=KG(3*MXYZ(g,1)-1,3*MXYZ(g,1)-1)*1.0e25;
        vddl(ad)=3*MXYZ(g,1)-1;
        ad=ad+1;
    end
    if MXYZ(g,5)==001
        KG(3*MXYZ(g,1),3*MXYZ(g,1))=KG(3*MXYZ(g,1),3*MXYZ(g,1))*1.0e25;
        vddl(ad)=3*MXYZ(g,1);
        ad=ad+1;
    end ; end;
    DvD = KG\FG;
MDD=zeros(NNOD,4);
fact=1; % facteur d'amplification
for i=1:NNOD
    MDD(i,1)=MXYZ(i,1);

```

```

MDD(i,2)=MXYZ(i,2)+(fact*depl(3*i-2));
MDD(i,3)=MXYZ(i,3)+(fact*depl(3*i-1));
MDD(i,4)=MXYZ(i,4)+(fact*depl(3*i));
end
for p = 1:NELEM
    E=MPRO(ME(p,22),2);
    Ro=MPRO(ME(p,22),3);
    v=MPRO(ME(p,22),4);
    E8N(p,:); % position matrice adresse des noeuds des element
    for i=1:20
        VDEC(3*i-2)=depl(3*(E8N(p,i))-2);
        VDEC(3*i-1)=depl(3*(E8N(p,i))-1);
        VDEC(3*i)=depl(3*(E8N(p,i)));
    end
    trans_vdec=VDEC';
    ir=ir; ngp=ir*ir*ir; nbpg=ngp*3;
    [gp,w]=Apl_Points_Gauss(ir);
    wp=w(:,1).*w(:,2).*w(:,3);
    ksi=gp(:,1); eta=gp(:,2); zeta=gp(:,3);
    [Nn,dNr]=Apl_Fonct_Forme(nbpg,ksi,eta,zeta);
    ed=VDEC; vect_vedt=VDEC;
    [XY]=Apl_VecXYZ(MXYZ,ME,p);
    ex=XY(:,1)'; ey=XY(:,2)'; ez=XY(:,3)';
    rowed=size(ed,1);
    incie=1;
    es=[]; et=[]; eci=[];
    ie=1;
    for ied=1:rowed
        eci=[eci Nn*[ex(1,:);ey(1,:);ez(1,:)]'];
        JT=dNr*[ex(1,:);ey(1,:);ez(1,:)]';
        Matrice_D
        for i=1:ngp
            indx=[ 3*i-2; 3*i-1; 3*i ];
            detJ=det(JT(indx,:));
            JTinv=inv(JT(indx,:));
            dNx=JTinv*dNr(indx,:);
            B(1,1:3:60-2)=dNx(1,:);
            B(2,2:3:60-1)=dNx(2,:);
            B(3,3:3:60) =dNx(3,:);
            B(4,1:3:60-2)=dNx(2,:);
            B(4,2:3:60-1)=dNx(1,:);
            B(5,1:3:60-2)=dNx(3,:);
            B(5,3:3:60) =dNx(1,:);
            B(6,2:3:60-1)=dNx(3,:);
            B(6,3:3:60) =dNx(2,:);
            ee=B*ed(ied,:)'; et=[et; ee']; es=[es; (D*ee)'];
        end; ie=ie+incie; end; end ;
% disp(' Contraintes aux points de Gauss du CENTRE ducube ');
% disp('-----');
% disp(' SigX SigY SigZ SigXY SigYZ SigXZ -----');
for p = 1:NELEM
    E8N(p,:); % position matrice adresse des noeuds des element
    for i=1:20
        VDEC(3*i-2)=depl(3*(E8N(p,i))-2);
        VDEC(3*i-1)=depl(3*(E8N(p,i))-1);
        VDEC(3*i)=depl(3*(E8N(p,i)));
    end
    trans_vdec=VDEC';
    irc=1; ngp=irc*irc*irc; nbpg=ngp*3;
    [gp,w]=Apl_Points_Gauss(irc);
    wp=w(:,1).*w(:,2);
    ksi=gp(:,1); eta=gp(:,1); zeta=gp(:,1);
    [Nn,dNr]=Apl_Fonct_Forme(nbpg,ksi,eta,zeta);
    ed=VDEC;
    vect_vedt=VDEC;
    [XY]=Apl_VecXYZ(MXYZ,ME,p);
    ex=XY(:,1)';
    ey=XY(:,2)';

```

```

ez=XY(:,3)';
incie=1;
es=[]; et=[]; eci=[];
ie=1;
E =MPRO(ME(p,22),2);
Ro =MPRO(ME(p,22),3);
v =MPRO(ME(p,22),4);
for ied=1: size(ed,1);
    eci=[eci Nn*[ex(1,:);ey(1,:);ez(1,:)]'];
    JT=dNr*[ex(1,:);ey(1,:);ez(1,:)]';
    Matrice_D
    for i=1:ngp
        indx=[ 3*i-2; 3*i-1; 3*i ];
        detJ=det(JT(indx,:));
        JTinv=inv(JT(indx,:));
        dNx=JTinv*dNr(indx,:);
        B(1,1:3:60-2)=dNx(1,:);
        B(2,2:3:60-1)=dNx(2,:);
        B(3,3:3:60) =dNx(3,:);
        B(4,1:3:60-2)=dNx(2,:);
        B(4,2:3:60-1)=dNx(1,:);
        B(5,1:3:60-2)=dNx(3,:);
        B(5,3:3:60) =dNx(1,:);
        B(6,2:3:60-1)=dNx(3,:);
        B(6,3:3:60) =dNx(2,:);
        ee=B*ed(ied,:)';
        et=[et; ee'];
        es=[es; (D*ee)'];
    end
    ie=ie+incie;
end
Mat_CPes(p,:)=es;
coodC=[MXYZ(E8N(p,4),2) MXYZ(E8N(p,9),3) MXYZ(E8N(p,2),4)];
cood_centre(p,:)=coodC;
end
File_result=[cood_centre Mat_CPes]

```

Apl_Fonct_Forme_cub20

function [Nn, dNr] = Apl_Fonct_Forme_cub20 (nbpq, ksi, eta, zeta)

```

Nn(:,1) = ((1-ksi) .* (1-eta) .* (1-zeta) .* (-ksi-eta-zeta-2))/8;
Nn(:,2) = ((1-ksi) .* (1-eta) .* (1-zeta.^2))/4;
Nn(:,3) = ((1-ksi) .* (1-eta) .* (1+zeta) .* (-ksi-eta+zeta-2))/8;
Nn(:,4) = ((1-ksi).^2) .* (1-eta) .* (1+zeta))/4;
Nn(:,5) = ((1+ksi) .* (1-eta) .* (1+zeta) .* (ksi-eta+zeta-2))/8;
Nn(:,6) = ((1+ksi) .* (1-eta) .* (1-zeta.^2))/4;
Nn(:,7) = ((1+ksi) .* (1-eta) .* (1-zeta) .* (ksi-eta-zeta-2))/8;
Nn(:,8) = ((1-ksi).^2) .* (1-eta) .* (1-zeta))/4;
Nn(:,9) = ((1-ksi) .* (1-eta).^2) .* (1-zeta))/4;
Nn(:,10) = ((1-ksi) .* (1-eta).^2) .* (1+zeta))/4;
Nn(:,11) = ((1+ksi) .* (1-eta).^2) .* (1+zeta))/4;
Nn(:,12) = ((1+ksi) .* (1-eta).^2) .* (1-zeta))/4;
Nn(:,13) = ((1-ksi) .* (1+eta) .* (1-zeta) .* (-ksi+eta-zeta-2))/8;
Nn(:,14) = ((1-ksi) .* (1+eta) .* (1-zeta.^2))/4; %---erreur corrigée
Nn(:,15) = ((1-ksi) .* (1+eta) .* (1+zeta) .* (-ksi+eta+zeta-2))/8;
Nn(:,16) = ((1-ksi).^2) .* (1+eta) .* (1+zeta))/4;
Nn(:,17) = ((1+ksi) .* (1+eta) .* (1+zeta) .* (ksi+eta+zeta-2))/8;
Nn(:,18) = ((1+ksi) .* (1+eta) .* (1-zeta.^2))/4; %---erreur corrigée
Nn(:,19) = ((1+ksi) .* (1+eta) .* (1-zeta) .* (ksi+eta-zeta-2))/8;
Nn(:,20) = ((1-ksi).^2) .* (1+eta) .* (1-zeta))/4;
dNr(1:3:nbpq,1) = ((eta - 1) .* (zeta - 1) .* (eta + 2*ksi + zeta + 1))/8;
dNr(1:3:nbpq,2) = -((zeta.^2 - 1) .* (eta - 1))/4;
dNr(1:3:nbpq,3) = -((eta - 1) .* (zeta + 1) .* (eta + 2*ksi - zeta + 1))/8;
dNr(1:3:nbpq,4) = (ksi .* (eta - 1) .* (zeta + 1))/2;
dNr(1:3:nbpq,5) = ((eta - 1) .* (zeta + 1) .* (eta - 2*ksi - zeta + 1))/8;
dNr(1:3:nbpq,6) = ((zeta.^2 - 1) .* (eta - 1))/4;
dNr(1:3:nbpq,7) = -((eta - 1) .* (zeta - 1) .* (eta - 2*ksi + zeta + 1))/8;
dNr(1:3:nbpq,8) = -(ksi .* (eta - 1) .* (zeta - 1))/2;
dNr(1:3:nbpq,9) = -((eta.^2 - 1) .* (zeta - 1))/4;
dNr(1:3:nbpq,10) = ((eta.^2 - 1) .* (zeta + 1))/4;
dNr(1:3:nbpq,11) = -((eta.^2 - 1) .* (zeta + 1))/4;
dNr(1:3:nbpq,12) = ((eta.^2 - 1) .* (zeta - 1))/4;
dNr(1:3:nbpq,13) = -((eta + 1) .* (zeta - 1) .* (2*ksi - eta + zeta + 1))/8;
dNr(1:3:nbpq,14) = ((zeta.^2 - 1) .* (eta + 1))/4;
dNr(1:3:nbpq,15) = -((eta + 1) .* (zeta + 1) .* (eta - 2*ksi + zeta - 1))/8;
dNr(1:3:nbpq,16) = -(ksi .* (eta + 1) .* (zeta + 1))/2;
dNr(1:3:nbpq,17) = ((eta + 1) .* (zeta + 1) .* (eta + 2*ksi + zeta - 1))/8;
dNr(1:3:nbpq,18) = -((zeta.^2 - 1) .* (eta + 1))/4;
dNr(1:3:nbpq,19) = -((eta + 1) .* (zeta - 1) .* (eta + 2*ksi - zeta - 1))/8;
dNr(1:3:nbpq,20) = (ksi .* (eta + 1) .* (zeta - 1))/2;
dNr(2:3:nbpq+1,1) = ((ksi - 1) .* (zeta - 1) .* (2*eta + ksi + zeta + 1))/8;
dNr(2:3:nbpq+1,2) = -(ksi/4 - 1/4) .* (zeta.^2 - 1);
dNr(2:3:nbpq+1,3) = -((ksi - 1) .* (zeta + 1) .* (2*eta + ksi - zeta + 1))/8;
dNr(2:3:nbpq+1,4) = (ksi.^2/4 - 1/4) .* (zeta + 1);
dNr(2:3:nbpq+1,5) = ((ksi + 1) .* (zeta + 1) .* (2*eta - ksi - zeta + 1))/8;
dNr(2:3:nbpq+1,6) = (ksi/4 + 1/4) .* (zeta.^2 - 1);
dNr(2:3:nbpq+1,7) = -((ksi + 1) .* (zeta - 1) .* (2*eta - ksi + zeta + 1))/8;
dNr(2:3:nbpq+1,8) = -(ksi.^2/4 - 1/4) .* (zeta - 1);
dNr(2:3:nbpq+1,9) = -2*eta .* (ksi/4 - 1/4) .* (zeta - 1);
dNr(2:3:nbpq+1,10) = 2*eta .* (ksi/4 - 1/4) .* (zeta + 1);
dNr(2:3:nbpq+1,11) = -2*eta .* (ksi/4 + 1/4) .* (zeta + 1);
dNr(2:3:nbpq+1,12) = 2*eta .* (ksi/4 + 1/4) .* (zeta - 1);
dNr(2:3:nbpq+1,13) = -((ksi - 1) .* (zeta - 1) .* (ksi - 2*eta + zeta + 1))/8;
dNr(2:3:nbpq+1,14) = +(ksi/4 - 1/4) .* (zeta.^2 - 1);
dNr(2:3:nbpq+1,15) = -((ksi - 1) .* (zeta + 1) .* (2*eta - ksi + zeta - 1))/8;
dNr(2:3:nbpq+1,16) = -(ksi.^2/4 - 1/4) .* (zeta + 1);
dNr(2:3:nbpq+1,17) = ((ksi + 1) .* (zeta + 1) .* (2*eta + ksi + zeta - 1))/8;
dNr(2:3:nbpq+1,18) = -(ksi/4 + 1/4) .* (zeta.^2 - 1);
dNr(2:3:nbpq+1,19) = -((ksi + 1) .* (zeta - 1) .* (2*eta + ksi - zeta - 1))/8;
dNr(2:3:nbpq+1,20) = (ksi.^2/4 - 1/4) .* (zeta - 1);
dNr(3:3:nbpq+2,1) = ((eta - 1) .* (ksi - 1) .* (eta + ksi + 2*zeta + 1))/8;
dNr(3:3:nbpq+2,2) = -2*zeta .* (ksi/4 - 1/4) .* (eta - 1);
dNr(3:3:nbpq+2,3) = -((eta - 1) .* (ksi - 1) .* (eta + ksi - 2*zeta + 1))/8;
dNr(3:3:nbpq+2,4) = (ksi.^2/4 - 1/4) .* (eta - 1);

```

```

dNr(3:3:nbpg+2,5) = ((eta - 1).*(ksi + 1).*(eta - ksi - 2*zeta + 1))/8;
dNr(3:3:nbpg+2,6) = 2*zeta.*(ksi/4 + 1/4).*(eta - 1);
dNr(3:3:nbpg+2,7) = -((eta - 1).*(ksi + 1).*(eta - ksi + 2*zeta + 1))/8;
dNr(3:3:nbpg+2,8) = -(ksi.^2/4 - 1/4).*(eta - 1);
dNr(3:3:nbpg+2,9) = -(eta.^2 - 1).*(ksi/4 - 1/4);
dNr(3:3:nbpg+2,10) = (eta.^2 - 1).*(ksi/4 - 1/4);
dNr(3:3:nbpg+2,11) = -(eta.^2 - 1).*(ksi/4 + 1/4);
dNr(3:3:nbpg+2,12) = (eta.^2 - 1).*(ksi/4 + 1/4);
dNr(3:3:nbpg+2,13) = -((eta + 1).*(ksi - 1).*(ksi - eta + 2*zeta + 1))/8;
dNr(3:3:nbpg+2,14) = 2*zeta.*(ksi/4 - 1/4).*(eta + 1);
dNr(3:3:nbpg+2,15) = -((eta + 1).*(ksi - 1).*(eta - ksi + 2*zeta - 1))/8;
dNr(3:3:nbpg+2,16) = -(ksi.^2/4 - 1/4).*(eta + 1);
dNr(3:3:nbpg+2,17) = ((eta + 1).*(ksi + 1).*(eta + ksi + 2*zeta - 1))/8;
dNr(3:3:nbpg+2,18) = -2*zeta.*(ksi/4 + 1/4).*(eta + 1);
dNr(3:3:nbpg+2,19) = -((eta + 1).*(ksi + 1).*(eta + ksi - 2*zeta - 1))/8;
dNr(3:3:nbpg+2,20) = (ksi.^2/4 - 1/4).*(eta + 1);
end

```

Apl_Points_Gauss

```

function [gp,w]=Apl_Points_Gauss (ir)
if ir==1
    g1=0.0; w1=2.0;
    gp=[ g1 g1 ];
    w=[ w1 w1 ];
elseif ir==2
    g1=0.577350269189626; w1=1;
    gp(:,1)=[-1; 1; 1; -1; -1; 1; 1; -1];
    gp(:,2)=[-1; -1; 1; 1; -1; -1; 1; 1];
    gp(:,3)=[-1; -1; -1; -1; 1; 1; 1; 1];
    w(:,1)=[ 1; 1; 1; 1; 1; 1; 1; 1]*w1;
    w(:,2)=[ 1; 1; 1; 1; 1; 1; 1; 1]*w1;
    w(:,3)=[ 1; 1; 1; 1; 1; 1; 1; 1]*w1;
elseif ir==3
    g1=0.774596669241483;
    g2=0.;
    w1=0.5555555555555555;
    w2=0.8888888888888888;
    I1=[-1; 0; 1; -1; 0; 1; -1; 0; 1]';
    I2=[ 0; -1; 0; 0; 1; 0; 0; 1; 0]';
    gp(:,1)=[I1 I1 I1]'*g1;
    gp(:,1)=[I2 I2 I2]'*g2+gp(:,1);
    I1=abs(I1); I2=abs(I2);
    w(:,1)=[I1 I1 I1]'*w1;
    w(:,1)=[I2 I2 I2]'*w2+w(:,1);
    I1=[-1; -1; -1; 0; 0; 0; 1; 1; 1]';
    I2=[ 0; 0; 0; 1; 1; 1; 0; 0; 0]';
    gp(:,2)=[I1 I1 I1]'*g1;
    gp(:,2)=[I2 I2 I2]'*g2+gp(:,2);
    I1=abs(I1); I2=abs(I2);
    w(:,2)=[I1 I1 I1]'*w1;
    w(:,2)=[I2 I2 I2]'*w2+w(:,2);
    I1=[-1; -1; -1; -1; -1; -1; -1; -1];
    I2=[ 0; 0; 0; 0; 0; 0; 0; 0];
    I3=abs(I1);
    gp(:,3)=[I1 I2 I3]'*g1;
    gp(:,3)=[I2 I3 I2]'*g2+gp(:,3);
    w(:,3)=[I3 I2 I3]'*w1;
    w(:,3)=[I2 I3 I2]'*w2+w(:,3);
end ; end

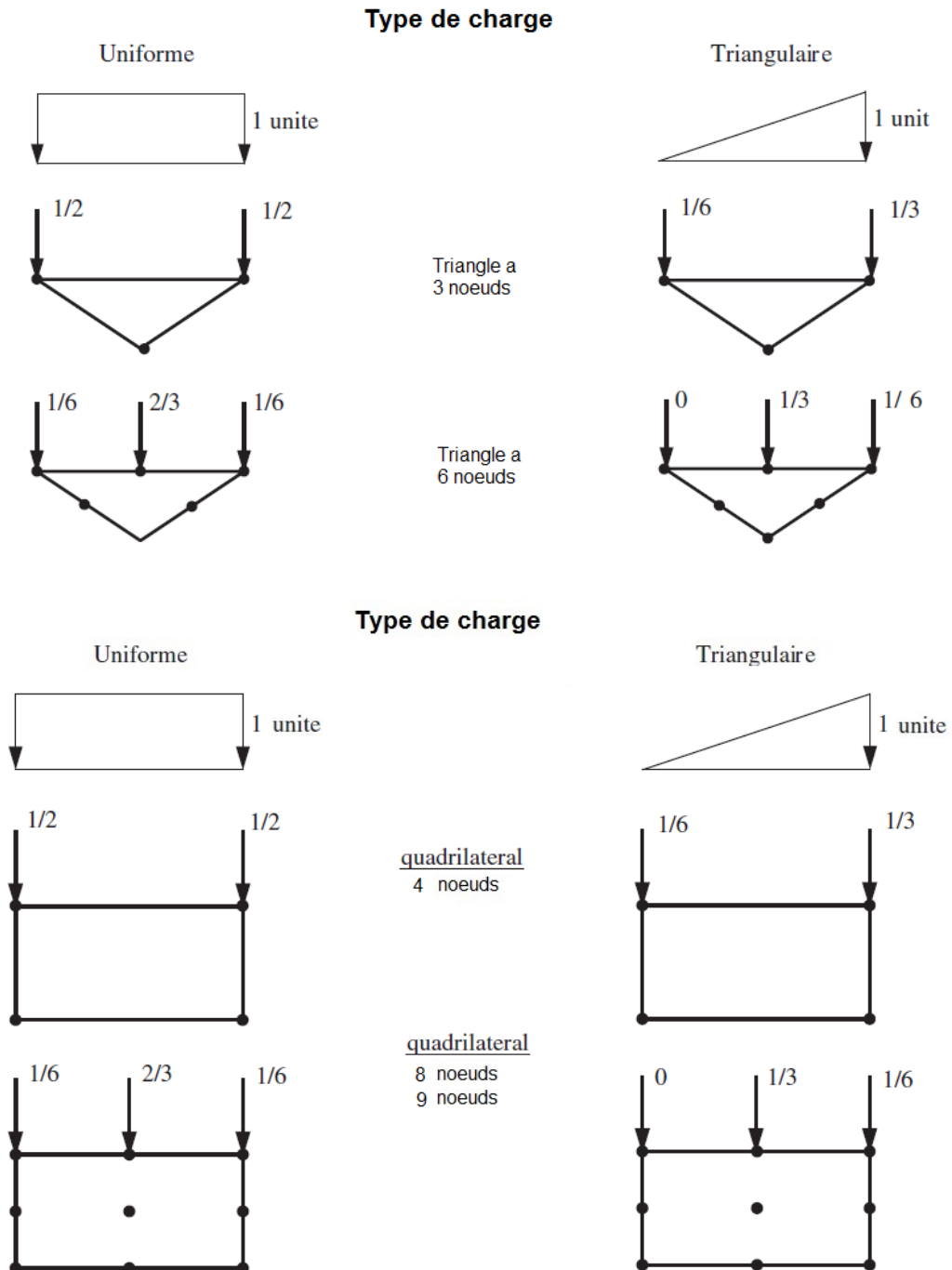
```

Annexe B

Chargement nodal équivalent

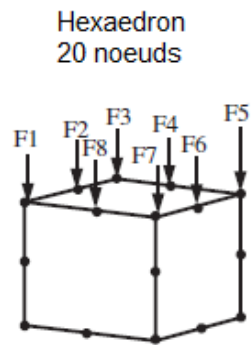
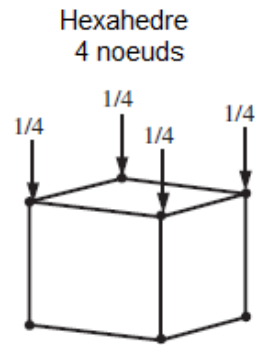
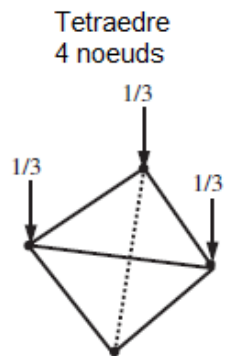
Éléments Plan (2D)

Avec un chargement unitaire



Éléments 3 Dimensions (3D)

Avec un chargement unitaire



$$F_1 = F_3 = F_5 = F_7 = -\frac{1}{12}$$

$$F_2 = F_4 = F_6 = F_8 = \frac{1}{3}$$

Table des figures

Figure 1 Principaux types d'éléments utilisé en 2D	33
Figure 02 : Principaux types d'éléments utilisé en 3D	34
Figure 3 : type d'éléments étudiés 1D.....	37
Figure 4 : Type d'éléments étudiés 2D.....	37
Figure 5 : Type d'éléments étudiés 3D.....	37
Figure 6 : élément Barre.....	40
Figure 7 : Elément Poutre.....	41
Figure 8 : Nœuds élément triangulaire T6	42
Figure 9 : Coordonnées élément quadratique à huit nœuds	44
Figure 10 : Coordonnées élément tétraèdre à 4 nœuds.....	48
Figure 11 Coordonnées élément cubique à 8 nœuds	50
Figure 12 : Fonction de forme cube 8 nœuds	50
Figure 13 : Hexaedron a 20 nœuds	54
Figure 14 : fonction de forme Hexaedron a 20 nœuds	55
Figure 15 : Intégration 27 points de Gauss Hexaèdre	64
Figure 16 : Intégration 8 points de Gauss Hexaèdre	64
Figure 17 : interface principale du programme Apl design_Stutures MARS <small>version 1.0</small>	76
Figure 18 : Interface Générateur de Maillage pour différents élément 2D	86
Figure 19 : forme circulaire maillée avec un élément rectangulaire Q4.....	87
Figure 20 : forme circulaire maillée avec un élément triangle T3.....	87
Figure 21 : forme rectangulaire avec trou circulaire au centre	88
Figure 22 : Elément quadratique iso-paramétrique Q8.....	88
Figure 23 : Interface générateur de Maillage pour élément tétraèdre T4	89
Figure 24 : Exemple Maillage Tétraèdre T4	89
Figure 25 : Interface générateur de Maillage pour élément Hexaèdre a 8 nœuds	91
Figure 26 : affichage maillage de la structure	91
Figure 27 : Interface générateur de maillage Hexaèdre 20 Nœuds.....	93
Figure 28 : Affichage du maillage de la structure.....	93
Figure 29 : L'interface du programme Apl_Design Barre.....	95
Figure 30 : Affichage de la structure	96
Figure 31 : Affichage des déplacements et des résultats.....	96
Figure 32 : interface Apl_design_Poutre.....	101
Figure 33 d : éformée de la structure	102
Figure 34 : Affichage Diagramme des efforts normaux	102
Figure 35 : Diagramme du moments fléchissant Affichage.....	103
Figure 36 : Affichage Diagramme des efforts tranchants.....	103
Figure 37 : Affichage du spectre de réponse.....	104
Figure 38 : Déformée animée de la structure durant un séisme	104
Figure 39 : Interface Apl_Design_triangle T6.....	108
Figure 40 : lancement des calculs	108

Figure 41 : animation des déplacements	109
Figure 42 : déplacement U_x	109
Figure 43 : Contrainte σ_x	109
Figure 44 : Contrainte τ_{xy}	109
Figure 45 : interface de Apl_design_Q8	112
Figure 46 : interface du module déplacements	112
Figure 47 : interface du Module contraintes	113
Figure 48 affichage d'un seul élément avec surbrillance son déplacement	114
Figure 49 affichage des contrainte et déformation aux point de Gauss de l'élément	114
Figure 50 : interface de Apl_Design_tetraedre	117
Figure 51 : interface du Module déplacements	117
Figure 52 : interface du Module contraintes	118
Figure 53 : Affichage matrice de rigidité	118
Figure 54 : Affichage des points de Gauss.....	118
Figure 55 Interface Apl_Design Cub8.....	121
Figure 56 Affichage des déplacements	121
Figure 57 Interface des déplacements	122
Figure 58 Interface des contraintes	122
Figure 59 :interface Apl_Design_Cub20.....	126
Figure 60 : Interface des déplacements	126
Figure 61 :Interface des contraintes	127
Figure 62 :Affichage matrice de rigidité globale	127
Figure 63 :Affiche 8 Points de Gauss	127
Figure 64 :Affiche 27 Points de Gauss.....	127
Figure 65 : exemple treillis	133
Figure 66 : affichage exemple treillis dans le programme	133
Figure 67 : déplacements dans la structure treillis	134
Figure 68 : résultats des calculs.....	134
Figure 69 : résultats obtenus avec Autodesk Robot Structural Analysis Professional.....	135
Figure 70 : Résultat obtenues avec prise en compte du poids propre	136
Figure 71 : Résultat obtenus avec Autodesk Robot Structural.....	136
Figure 72 : exemple portique	137
Figure 73 : affichage de la structure dans le programme Apl_design Poutre.....	137
Figure 74 : Déformée de la structure	138
Figure 75 : Résultat obtenus par Autodesk Robot Structural Analysis Professional.....	138
Figure 76 : exemple test 01 Poutre consol.....	139
Figure 77 : maillage de la poutre avec Apl_design_Q8	141
Figure 78 : résultats des déplacements	141
Figure 79 : Déformée et variation des contraintes par itération	142
Figure 80 : Distribution des contraintes σ_x dans une poutre	142
Figure 81 : Distribution des contraintes σ_y dans une poutre	143
Figure 82 : Distribution des contraintes σ_{xy} dans une poutre	143
Figure 83 : exemple test02.....	144
Figure 84 : Affichage maillage de la structure avec conditions aux limites	144
Figure 85 : Déformée de la structure avec un facteur d'échelle de 10.....	145
Figure 86 ; Distribution des contraintes σ_y dans une poutre.....	145
Figure 87 : variation de la déformée suivant U_y	146
Figure 88 : résultats des déplacements	146

Figure 89 : exemple test 01 Poutre consol.....	147
Figure 90 : maillage de la poutre avec Apl_design T6.....	147
Figure 91 : Déformée de la structure	148
Figure 92 : résultats des déplacements	148
Figure 93 : Déplacement et variation des contraintes.....	149
Figure 94 : Distribution des contraintes σ_x	149
Figure 95 : Distribution des contraintes σ_y	150
Figure 96 : Distribution des contraintes σ_{xy}	150
Figure 97 : Distribution des contraintes σ principale 1	151
Figure 98 : Distribution des contraintes σ principale 2	151
Figure 99 : Résultats des déplacements.....	152
Figure 100 : exemple test 01 Poutre consol.....	153
Figure 101 : maillage de la structure.....	153
Figure 102 :Déformée de la structure	154
Figure 103 : résultats des déplacement	154
Figure 104 : distribution des contraintes σ_y	155
Figure 105 : résultat des contraintes	155
Figure 106 : exemple test 02 cube tétraèdre.....	156
Figure 107 : maillage de cube tetraedre	156
Figure 108 : Résultats obtenus.....	157
Figure 109 : résultats données par Griffiths.....	157
Figure 110 : exemple test 01 Poutre consol.....	158
Figure 111 : maillage de la structure.....	158
Figure 112 : déformée de la structure	159
Figure 113 : résultats des déplacements	159
Figure 114 : distribution des contraintes σ_y	160
Figure 115 : distribution des contraintes σ_z	160
Figure 116 : distribution des contraintes σ principale 1	161
Figure 117 : résultats contraintes	161
Figure 118 : distribution des contraintes σ principale 2	162
Figure 119 : distribution des contraintes σ principale 3	162
Figure 120 : exemple test 02.....	163
Figure 121 : maillage de la structure.....	163
Figure 122 : maillage de structure et affichage des résultats	164
Figure 123 : déformée de la structure	164
Figure 124 distribution des contraintes σ_y	165
Figure 125 : Résultats des déplacements.....	165
Figure 126 : résultats de comparaison.....	165
Figure 127 : exemple test 01 Poutre consol.....	166
Figure 128 : maillage de la structure avec Apl_mesh_tetredre	166
Figure 129 : Affichage de la structure	167
Figure 130 déformée de la structure	167
Figure 131 Résultats des déplacements.....	168
Figure 132 : distribution des contraintes	168
Figure 133 : résultats des contraintes.....	169
Figure 134 : distribution des contraintes σ_z	170
Figure 135 : distribution des contraintes σ_x	170
Figure 136 exemple test2.....	171

Figure 137 maillage et affichage de la structure.....	171
Figure 138 résultats des déplacements.....	172
Figure 139 : résultats des contraintes.....	172
Figure 140 : résultats de comparaison.....	172
Figure 141 : l'évolution du déplacement vertical w le long de la poutre	173
Figure 142 : évolution du déplacement le long de la poutre pour différents nombre de division par d'élément Q8	175
Figure 143 : évolution du déplacement le long de la poutre pour différents nombre de division par d'élément T6	176
Figure 144 : évolution du déplacement le long de la poutre pour différents nombre de division par d'élément T4	176
Figure 145 : évolution du déplacement le long de la poutre pour différents nombre de division par d'élément C8.....	177
Figure 146 : évolution du déplacement le long de la poutre pour différents nombre de division par d'élément C 20.....	178
Figure 147 modélisation de la demi pile.....	180
Figure 148 : exemple 01 pile de pont.....	181
Figure 149 : affichage de la structure.....	181
Figure 150 : résultats de déplacements et contraintes	182
Figure 151 : déformée de la structure Ux avec un facteur d'échelle.....	183
Figure 152 : déplacement Uy avec un facteur d'échelle.....	183
Figure 153 : distribution des contraintes σ_y	184
Figure 154 : distribution des contraintes σ_x	184
Figure 155 : distribution des contraintes $\sigma_{\text{principale 1}}$	185
Figure 156 : distribution des contraintes $\sigma_{\text{principale 2}}$	185
Figure 157 : maillage structure plaque trouée.....	186
Figure 158 : déformée de la structure avec un facteur d'échelle	187
Figure 159 : résultats des déplacements et des contraintes	187
Figure 160 : distribution des contraintes σ_x	188
Figure 161 : distribution des contraintes σ_y	188
Figure 162 : distribution des contraintes σ_{xy}	189
Figure 163 : structure tunnel	190
Figure 164 : maillage de la structure.....	190
Figure 165 : déformée de la structure avec un facteur d'échelle	191
Figure 166 : déformée de la structure Ux	191
Figure 167 déformée de la structure Uy	192
Figure 168 : distribution des contraintes σ_y	192
Figure 169 : distribution des contraintes σ_{xy}	193
Figure 170 : distribution des contraintes de Von Mises	193
Figure 171 exemple rideau palplanche	194
Figure 172 : maillage de l'exemple a étudier	194
Figure 173 : déformée du sol sans Palplanche	195
Figure 174 : déformée du sol sans Palplanche	195
Figure 175 : déplacements et déformée du sol sans Palplanche.....	196
Figure 176 : déplacements et déformée du sol avec Palplanche.....	196
Figure 177 : distribution des contraintes σ_{xy} σ_y sans Palplanche	197
Figure 178 distribution des contraintes σ_{xy} σ_y avec Palplanche	197
Figure 179 : distribution des contraintes Principale σ_1 σ_2	198

Figure 180 : distribution des contraintes Principale σ_2 et de von mises	198
Figure 181.....	199
Figure 182 : maillage de la structure.....	199
Figure 183 : Maillage et conditions aux limites.....	200
Figure 184 : déformée de la structure	200
Figure 185 : résultats déplacements	201
Figure 186 : déplacement suivant Uy.....	201
Figure 187 : déplacement suivant Ux.....	202
Figure 188 distribution des contraintes σ_x	202
Figure 189 distribution des contraintes σ_{xy}	203
Figure 190 distribution des contraintes σ_z	203
Figure 191 distribution des contraintes $\sigma_{\text{von mises}}$	203
Figure 192 exemple 02	204
Figure 193 : maillage de la structure.....	204
Figure 194 : Maillage et conditions aux limites.....	205
Figure 195 : déformée de la structure	205
Figure 196 : résultats déplacements	206
Figure 197 : déplacement suivant Uy.....	207
Figure 198 : déplacement suivant Ux.....	207
Figure 199 distribution des contraintes σ_x	208
Figure 200 distribution des contraintes σ_{xy}	208
Figure 201 distribution des contraintes σ_z	208
Figure 202 distribution des contraintes $\sigma_{\text{von mises}}$	208
Figure 203 : structure a étudier	209
Figure 204 : maillage et affichage de la structure dans le programme	209
Figure 205 : distribution des contraintes σ_x	210
Figure 206 : distribution des contraintes σ_y	210
Figure 207 : distribution des contraintes σ_y	211
Figure 208 : distribution des contraintes σ_y	211
Figure 209 : déformée de la structure avec un facteur d'échelle	212
Figure 210 : structure Poutre en H.....	213
Figure 211 : maillage de la structure.....	213
Figure 212 : déplacements de la structure Ux	214
Figure 213 : déformée Uy de la structure	214
Figure 214 : déformée Uz de la structure	215
Figure 215 : distribution des contraintes σ_z	215
Figure 216 : distribution des contraintes σ_{yz}	216
Figure 217 : distribution des contraintes $\sigma_{\text{principale 3}}$	216
Figure 218 : distribution des contraintes $\sigma_{\text{principale 2}}$	217
Figure 219 : distribution des contraintes $\sigma_{\text{principale 1}}$	217
Figure 220 : structure a étudier	218
Figure 221 : maillage de la structure.....	218
Figure 222 : distribution des contraintes σ_x	219
Figure 223 : distribution des contraintes σ_y	219
Figure 224 : distribution des contraintes σ_{xy}	220
Figure 225 : distribution des contraintes $\sigma_{\text{principale 2}}$	220
Figure 226 : structure à étudier semelle	221
Figure 227 : maillage de la structure avec conditions aux limites	221

Figure 228 : résultats des déplacements	222
Figure 229 ; matrice de rigidité	222
.....	Figure 230 : déformée Ux suivants x
.....	223
Figure 231 : déformée Uz suivant Z.....	223
Figure 232 : déformée Ux suivant x	224
Figure 233 : résultats des contraintes.....	224
Figure 234 : distribution des contraintes σ_x	225
Figure 235 : distribution des contraintes σ_z	225
Figure 236 : distribution des contraintes σ_y	226
Figure 237 : distribution des contraintes $\sigma_{\text{principale 3}}$	226
Figure 238 exemple 03.....	227
Figure 239 : maillage du terrain avec Apl_mesh hexaedron.....	227
Figure 240 : distribution des déplacement Ux.....	228
Figure 241 : distribution des déplacement Uz	228
Figure 242 distribution des contraintes σ_x	229
Figure 243: distribution des contraintes σ_y	229
Figure 244 distribution des contraintes σ_z	230
Figure 245: distribution des contraintes τ_{xy}	230

Liste des tableaux

Tableau 1 : Coordonnées des points de Gauss et poids dans les éléments rectangulaires.....	62
Tableau 2 : Coordonnées des points de Gauss et poids dans les éléments triangulaires	63
Tableau 3 : Intégration de Gauss pour l'éléments tétraèdre	63
Tableau 4 : Intégration avec 8 points de Gauss pour Hexaèdre	64
Tableau 5 : Intégration avec 27 points de Gauss pour Hexaèdre	65
Tableau 6 : sous programmes Apl_Disign_Barre	99
Tableau 7 : Subroutines Apl_design_Poutre.....	105
Tableau 8 : Sous-programmes Apl_Design_triangle T6	110
Tableau 9 :Subroutines Apl_Design_Quadratique Q8	115
Tableau 10 : Soubrogrammes Apl_Design_Tétraedre	119
Tableau 11 :Sous programmes Apl_Design_Cub8	123
Tableau 12 : Subroutines Apl_Design_Cub20	128
Tableau 13 valeurs comparatives des résultats	173
Tableau 14 Résultats RDM	175
Tableau 15 ; division élément T6	176

