MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE UNIVERSITE DR. YAHIA FARES DE MEDEA
FACULTE DES SCIENCES ET DE LA TECHNOLOGIE
Département du Génie



Projet de Synthèse Présenté pour l'obtention du
Diplôme De master 2 en Génie Civil
Option : Géotechnique

# MODELISATION PAR ELEMENTS FINIS D'UN MASSIF DE SOL SOUS LES SOLLICITATIONS VERTICALES

M. Belhacene Amine
M. Zemir Mohammed Abdelfettah

Proposé et dirigé par

Mr. Bakhti Rachid

# Remerciements

Nous remercions Dieu le Tout Puissant, qui nous a aidé et donné la force et la persévérance de mener à bien ce travail, notre reconnaissance envers Dieu est sans limite. Nos remerciements vont ensuite à : À notre encadreur,Bakhti Rachid qui nous a honoré en acceptant d'encadrer notre travail et qui par sa grande compréhension et judicieuses directives nous a été d'une grande aide, ainsi que par une disponibilité sans égal.Il nous a appris à affronter toutes les difficultés qui se sont posées à nous .Comme je le remercie également pour sa générosité et sa collaboration d'avoir corrige et examine notre travail.Mille fois merci Monsieur Bakhti Rachid.

Nous exprimons, aussi notre reconnaissance et notre gratitude aux membres du jury qui ont accepté d'évaluer notre travail en y joignant leurs temps, leurs savoirs et leurs expériences. Nous n'oublierons pas tous nos professeurs de l'université de YAHIA FARES département GENIE CIVIL qui nous ont donné le savoir et la formation requise.

À nos parents, père, mère, frères et sœurs qui ont supporté avec nous toutes les difficultés matérielles, financières et morales qui se sont posées à nous pendant la période de réalisation de ce travail. À tous ceux et celles qui nous ont apporté aide et soutien, nous apprécions toutes vos interventions et vous remercions du fond du cœur.

# Dédicaces

Que ce travail témoigne de notre respect :

A nos parents :

Grâce à leurs tendres encouragements et leurs grands sacrifices, ils ont pu créer le climat affectueux et propice à la poursuite de nos études. Aucune dédicace ne pourrait exprimer notre respect, notre considération et notre profond sentiment envers eux. Nous prions le bon Dieu de les bénir, de veiller sur eux, en espérant qu'ils seront toujours fiers de nous.

A nos sœurs et à nos frères.

A nos familles

Belhacene et zemir.

Ils vont trouver ici l'expression de notre sentiment de respect et de Reconnaissance pour le soutien qu'ils n'ont cessé de nous porter.

A tous nos professeurs leur générosité et leur soutien nous obligent à leurs Témoigner notre profond respect et notre loyale considération.

A tous nos amis et nos collègues spécialement Zakaria, Salah, Fouad et Madani, Ils vont trouver ici le témoignage d'une fidélité et d'une amitié infinie.

Amine et Abdelfettah

# Table des matières

**I Les contrainte dans le sol** **3**

I.1 Introduction . . . . . 3

I.2 Les types des contraintes . . . . . 3

I.2.1 Contrainte effective . . . . . 3

I.2.2 Contrainte réelle –principe de superposition . . . . . 4

I.2.3 Contraint naturelle au géostatique . . . . . 4

I.3 Méthodes analytique des calculs des contraintes dans le sol . . . . . 4

I.3.1 Les contraintes dans le sol dues aux charges extérieures . . . . . 4

I.4 Méthodes numérique pour le calcul des contraintes dans le sol . . . . . 9

I.4.1 Intérêt des méthodes numériques . . . . . 9

**II Méthode des éléments finis** **10**

II.1 Introduction . . . . . 10

II.1.1 L'idees de la MEF . . . . . 10

II.1.2 Equation fondamentale de la MEF . . . . . 11

II.1.3 Etapes logiques du calcul par éléments finis . . . . . 11

II.1.4 Type de problèmes élément finie . . . . . 11

II.1.5 Problèmes d'équilibre stationnaire . . . . . 11

II.1.6 Problèmes aux valeurs propres . . . . . 12

II.1.7 Problèmes dépendants du temps . . . . . 12

II.2 Discrétisation . . . . . 12

II.2.1 Fonction de forme . . . . . 12

II.2.2 Type d'éléments . . . . . 14

II.2.3 Fonction de forme D'élément a une dimension . . . . . 15
II.2.4 Les conditions aux nœuds sous la forme matricielle . . . . . 18
II.3 Constructions des fonctions de forme . . . . . 18
II.3.1 Polynômes de Lagrange . . . . . 21
II.3.2 Polynôme d 'Hermite . . . . . 23
II.3.3 Condition de conformité . . . . . 25
II.4 Eléments iso paramétriques . . . . . 25
II.5 Transformation géométrique . . . . . 25
II.5.1 Transformation d'un d élément linéique . . . . . 27
II.6 Matrice de rigidité . . . . . 28
II.6.1 Formulation de la matrice b . . . . . 29
II.6.2 MatriceD . . . . . 31
II.6.3 Forces élémentaires . . . . . 32
II.6.4 Assemblage des matrice de rigidité . . . . . 35

**III Les problemes axisymetrique** **37**
III.1 Introduction . . . . . 37
III.2 Elément finie choisi . . . . . 39
III.2.1 Les éléments quadrangulaires à 8 nœuds . . . . . 40
III.3 Matrice de rigidité . . . . . 41
III.3.1 Condition ou limite . . . . . 42
III.4 Les deplacement . . . . . 43
III.4.1 Cas 1-deplacement nul $u_i$ . . . . . 43
III.4.2 Cas 2- déplacement imposer non nul $u_i = \overline{u_i} \neq 0$ . . . . . 43
III.5 Les contraintes . . . . . 44

**IV Présentation de Language Visuel Basic 6** **45**
IV.1 Les principaux concepts de Visual Basic . . . . . 45
IV.1.1 Présentation générale . . . . . 45
IV.2 Le modèle événementiel . . . . . 46
IV.2.1 La notion d'objet . . . . . 46

IV.2.2 Les notions de méthode et de propriété . . . . . 47
IV.3 Prise en main de l'environnement . . . . . 47
IV.3.1 Démarrage de Visual Basic . . . . . 47
IV.3.2 Les éléments de l'environnement . . . . . 48
IV.4 Structure d'une application Visual Basic . . . . . 51
IV.4.1 La création d'une application . . . . . 53
IV.4.2 Création de l'interface . . . . . 54
IV.4.3 Écriture du code . . . . . 56
IV.4.4 Exécution de l'application . . . . . 59

**V Conception de S.U.A.S v1.0** **60**
V.1 les entrés . . . . . 61
V.1.1 Propriété géométrique . . . . . 61
V.1.2 Propriété des matériaux . . . . . 61
V.1.3 Propriété des maillages . . . . . 62
V.1.4 La charge . . . . . 63
V.1.5 Nombre de point de gauss . . . . . 63
V.2 les sortie . . . . . 63
V.2.1 Déplacement . . . . . 63
V.2.2 Déformation . . . . . 64
V.2.3 Les contraintes . . . . . 64
V.3 La programmation des éléments finis . . . . . 65
V.3.1 L'algorithme . . . . . 65

**VI Comparaison entre la méthode de boussinesq et SUAS v1.0** **69**
VI.1 Les exemples de calcules . . . . . 69
VI.2 Interprétationdes résultats . . . . . 79
VI.3 Conclusion générale . . . . . 79

**Appendices** **80**

**A Subroutune numérotation ELE et coordonnée coord** **81**

**B Subroutine matrice D** **83**

**C Subroutine de vecteur function de form FUN et matrice derive DER** **85**

**D Subroutine l'élimination de gausse** **87**

# Table des figures


I.1 Distribution de $\sigma_z$ on fonction de $\theta$ . . . . . 5
I.2 Charge linéaire uniforme répartie sur une longueur Infinie . . . . . 6
I.3 Charge Uniform linéaire . . . . . 6
I.4 Charge uniforme sur une bande de longueur infinie . . . . . 7
I.5 Charge uniformément repartie . . . . . 7
I.6 Charge uniforme repartie Sur une surface circulaire . . . . . 8

II.1 Diffèrent type élément finis . . . . . 13
II.2 déférente type d'élément fini 3D . . . . . 14
II.3 déférente type d'élément fini 2D . . . . . 15
II.4 fonctions de forme de l'élément À 2 nœuds . . . . . 16
II.5 fonctions de forme de l'élément à 3 nœuds . . . . . 17
II.6 Fonction de forme d'un élément de Lagrange a quatre nœuds . . . . . 22
II.7 Elément quadrangulaire à 4 nœuds . . . . . 24
II.8 Diffèrent types d'élément paramétrique . . . . . 26
II.9 transformations d'un élément linéique à trois nœuds . . . . . 27
II.10 Elément quadrangulaire à 8 NŒUDS . . . . . 29
II.11 méthode de trapèzes . . . . . 33
II.12 Méthodes de Simpson . . . . . 33
II.13 Méthodes de Gauss . . . . . 34

III.1 Déformations-déplacements-contraints . . . . . 38
III.2 Corps axisymétrique . . . . . 38
III.3 Elément choisi . . . . . 39

III.4 Eléments quadrangulaires à 8 nœuds . . . . . 40

IV.1 Fenetre Démarrage de Visual Basic . . . . . 48
IV.2 Les éléments de l'environnement . . . . . 49
IV.3 Feuille et son module de feuille (code correspondant) . . . . . 51
IV.4 Conception de froms . . . . . 53
IV.5 Controles . . . . . 54
IV.6 La fenêtre Propriétés . . . . . 55
IV.7 Les valeurs des controles . . . . . 56
IV.8 Les valeurs des controles . . . . . 57
IV.9 Code en VB 6.0 . . . . . 58

V.1 interface de programme SUAS v 1.0 . . . . . 60
V.2 18manuel utilisation SUAS v 1.0 . . . . . 61
V.3 Module d'élasticité . . . . . 61
V.4 Coefficient de poisson . . . . . 62
V.5 Paramètres des maillages . . . . . 62
V.6 La charge concentrée . . . . . 63
V.7 Nombre de point de gauss . . . . . 63
V.8 Déplacements vertical et horizontale . . . . . 64
V.9 les quatre déformations . . . . . 64
V.10 Les quatre contraintes . . . . . 65
V.11 numérotations locales et globales . . . . . 66

VI.1 Les resultats du S.U.A.S . . . . . 70
VI.2 Le graphe de contrainte en fonction de la profondeur . . . . . 70
VI.3 Les resultats du S.U.A.S . . . . . 71
VI.4 Le graphe de contrainte en fonction de la profondeur . . . . . 72
VI.5 Les resultats du S.U.A.S . . . . . 73
VI.6 Le graphe de contrainte en fonction de la profondeur . . . . . 73
VI.7 Les resultats du S.U.A.S . . . . . 74

VI.8 Le graphe de contrainte en fonction de la profondeur . . . . . 74
VI.9 Les resultats du S.U.A.S . . . . . 75
VI.10Le graphe de contrainte en fonction de la profondeur . . . . . 76
VI.11Les resultats du S.U.A.S . . . . . 77
VI.12Le graphe de contrainte en fonction de la profondeur . . . . . 77

# Liste des tableaux

II.1 Fonction de forme des éléments linéiquesà 2 ,3 et 4 nœuds . . . . . 18
II.2 Integration dans le sol . . . . . 35

III.1 Fonctions d'interpolations et leurs dérivées partielles . . . . . 41

VI.1 Résume des résultats calculés . . . . . 78
VI.2 Résume des résultats calculés . . . . . 78

# Introduction Générale

Pour mener à bien un projet géotechnique, l'ingénieur doit concevoir puis dimensionner l'ouvrage de manière à fournir à l'entreprise exécutants les plan nécessaire à sa réalisation .Très souvent se lui -ci utilisera un outil de calcul basé sur la méthode des éléments finis d'ont l'utilisation s'est généraliser dans l'industrie depuis les années soixante.

Cette méthode qui n'est pas uniquement dédiée aux problèmes de structure lui permettra de résoudre un éventail très large de problème en géotechniques avec des aspects linéaires ou non linéaires, stationnaire ou transitoire.

Défèrent éditeur de logiciel se sont imposé sur le marché .il propose généralement plusieurs modules permettant d'aborde des problèmes en génie civile de ces code comporte habituellement un préprocesseur un ou plusieurs solveurs ,un ou plusieurs post –processeur .le près processeur est une interface graphique permettant a l'utilisateur de décrire la géométrie et le type de problèmes à résoudre le ou les solveur intégrant les base des méthodes de résolution (linéaires ou non linéaires stationnaire ou transitoire ,etc.) spécifique au cas étudie les post processeur permettant de visualiser les résultats sous forme de courbe (évolutions en fonction du temps des charge des déplacements etc.)

Mais avant d'utiliser un tel Code de calcul de manière opérationnelle il est essentielle d'explore ses capacité et surtout ses limites .pour ce faire ,le future utilisateur devra maitriser un minimum de prérequis théorique dans le secteur vise génie civil mais également dans le domaine de la méthodes des élément finis toujours dans ce même domaine et au niveau pratique il devra être capable de résoudre des problèmes simple avec des logiciel mais à sa disposition généralement .

Les codes éléments finis font maintenant partie des outils couramment utilisés lors de la conception Et à l'analyse des problèmes en géotechnique. Les outils d'aide à la modélisation

devenant de plus en plus Perfectionnés, l'utilisation de la méthode des éléments finis s'est largement développée et peut sembler de Moins en moins une affaire de spécialistes. Si l'utilisation de la méthode se démocratise de par la Simplicité croissante de mise en œuvre, la fiabilité des algorithmes et la robustesse de la méthode, il reste Néanmoins des questions essentielles auxquelles l'ingénieur devra répondre s'il veut effectuer une analyse Par éléments finis dans de bonnes conditions. Il lui faudra :

— Formaliser les non-dits et les réflexions qui justifient les choix explicites ou implicites de son Analyse du problème.

— Évaluer la confiance qu'il accorde aux résultats produits.

— Analyser les conséquences de ces résultats par rapport aux objectifs visés.

L'objectif de cette partie est de présenter les principes de base de cette méthode en insistant sur L'enchaînement des tâches (démarche et hypothèses associées) qui assurent la cohérence du processus de Calcul. Ces connaissances vous seront utiles pour maîtriser les deux principales difficultés de mise au Point d'un modèle numérique :

— Problèmes préliminaires à la phase de calcul.

— Problèmes liés à l'exploitation des résultats et le retour à la conception.

Ne perdez jamais de vue que l'analyse des résultats nécessite une bonne compréhension des Différentes étapes mathématiques utilisées lors de l'approximation, pour pouvoir estimer l'erreur du modèle numérique par rapport à la solution exacte du problème mathématique. Sans oublier que le modèle numérique ne peut fournir que des résultats relatifs aux Informations contenues dans le modèle mathématique qui découle des hypothèses de modélisation. Nous nous limiterons à la présentation de modèles élémentaires utilisés dans le cadre des théories Linéaires. Bien que simples ces modèles permettent déjà de traiter un grand nombre d'applications liées Aux problèmes géotechniques. Du point de vue pédagogique, ils sont suffisamment complexes pour mettre En avant les difficultés de mise en œuvre de la méthode.

Dans notre travaille nous avons utilisé la méthode d'élément fini pour modéliser un sol et faire une comparaison de résultats trouver les contrainte et les compare avec les résultats de Boussinesq.

# Chapitre I

# Les contrainte dans le sol

## I.1 Introduction

La distributions des contrainte dans le sol dans le Domain de génie civil a un rôle très important en effet la stabilité des structures et des ouvrages se base sur la fiabilité du note calcul. Plusieurs méthodes ont été élaborées pour nous permettre de calculer avec une précision très abordable Les contraintes dans les littératures à titre d'exemple la méthode de Boussinesq et ...

Ces méthodes sont exploitable jusqu' à nos jours malgré le développement actuelle par ailleurs en trouve des méthodes qui se base sur des technique moderne par exemple la méthode des éléments finis Dans ce chapitre on va présenter en détail les méthodes analytiques talque la méthode de Boussinesq.

## I.2 Les types des contraintes

### I.2.1 Contrainte effective

L'idée de séparer les contraintes pour chaque phase revient à terzaghi qui a postule L'existence d'un nouveau tenseur des contraintes, le tenseur des contraintes effectives qui gouverne le comportement du squelette granulaire des sol.

### I.2.2 Contrainte réelle –principe de superposition

Le sol est assimile à un milieu semi –infini élastique a surface horizontale le calcul des contrainte dans un massif pesant et charger est base sur l'utilisation du principe e superposition ce dernier résulte de l'hypothèse sur la loi de comportement du sol : élastique linéaire.

La contrainte réelle s'exercent a la profondeur z sur une facette horizontale,$\sigma_z$est égalea la somme de la contrainte naturelle $\sigma_{z0}$ du au pois du sol sus jacent et de la contrainte due aux surcharges $\sigma_z = \sigma_{v0} + \Delta\sigma_z$

### I.2.3 Contraint naturelle au géostatique

On appelle contraint naturelle (au géostatique) la contrainte s'exercent dans le sol a surface libre horizontale, sur une facette horizontale, avant tout chargement elle correspond généralement au poids des terres qui surmontent le point considère la facette considérée ayant sa normale verticale, la contrainte normale correspondante est notée :$\sigma_{v0}$

## I.3 Méthodes analytique des calculs des contraintes dans le sol

### I.3.1 Les contraintes dans le sol dues aux charges extérieures

#### Introduction

Nous allons étudier les résultats des théories de calcul des contraintes dans le sol dues aux charges extérieures telles que poids des ouvrages les charges d'exploitation et poids des équipements sur chantier.

**Charge concentrée verticale problème en 3D (Problème de Boussinesq)** Les hypothèses de calcul sont

— Un domaine a 3D semi-infini

— Milieu élastique non pesant sans aucune force de volume isotrope et homogène

— Plan supérieur horizontal

— La charge extérieure est concentrée verticale

La contrainte verticale due à la charge extérieure concentrée est donnée par la solution de boussinesq :

$$\sigma_z = \frac{3p}{2\pi z^2}\cos^5\theta = \frac{3pz^3}{2\pi R^5} \tag{I.1}$$

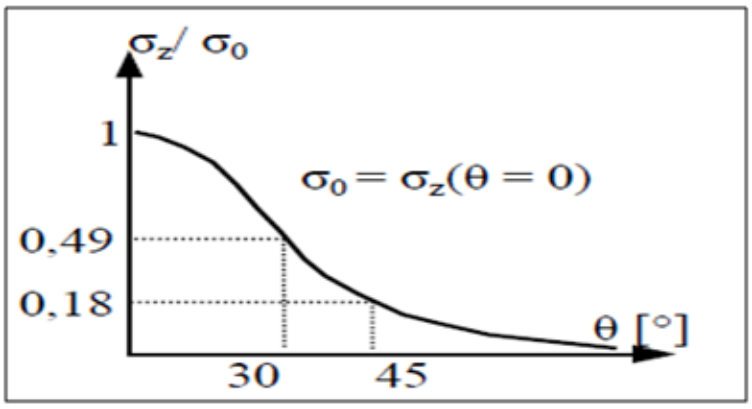



FIGURE I.1 – Distribution de $\sigma_z$ on fonction de $\theta$

On remarque que la contrainte $\sigma_z$ est indépendante des propriétés du massif de sol. Pour le calcul rapide de la contrainte, on peut utiliser un abaque approprié La contrainte verticale dans le sol vari avec la position dans le sol du point de calcul Les courbes d'égale pression sont comme indiquer ci-contre.

Calcul par abaque de la contrainte transmise au sol par une charge extérieure concentrée (d'après introduction à la géotechnique). NB pour la théorie de Boussinesq, NW pour la théorie de Westerwald

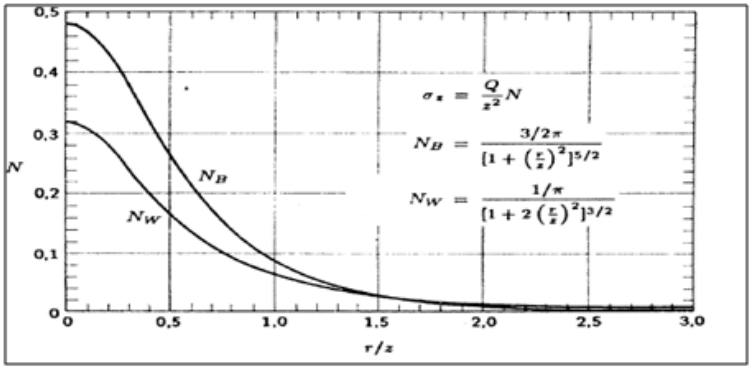



FIGURE I.2 – Charge linéaire uniforme répartie sur une longueur Infinie

La solution est due à flamant :

$$\sigma_z = \frac{2q}{\pi R}\cos^3\theta \tag{I.2}$$

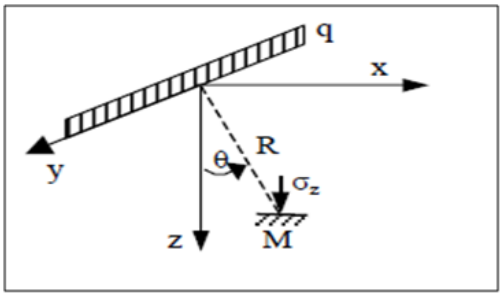



FIGURE I.3 – Charge Uniform linéaire

**Charge uniforme répartie sur une bande de longueur infinie et de largeur finie**

La bande se présente comme indiquer ci-contre On basant sur la solution de Flamant, on obtient l'expression :

$$\sigma_z = \frac{q}{z}[(\theta_2 - \theta_1) + \frac{\sin 2\theta_2 - \sin 2\theta_1}{2}] \tag{I.3}$$

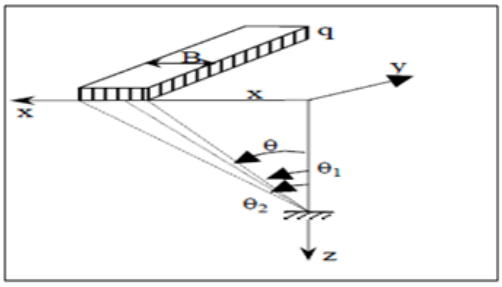



FIGURE I.4 – Charge uniforme sur une bande de longueur infinie

### Charge uniformément répartie

Soit q l'intensité de la charge répartie La solution est basé sur la solution de Boussinesq :

$$\sigma_z = \frac{3p}{2\pi z^2} \int_\alpha \cos^5 \theta d.A \tag{I.4}$$

La contrainte dépend de l'aire A de répartition de la charge q.

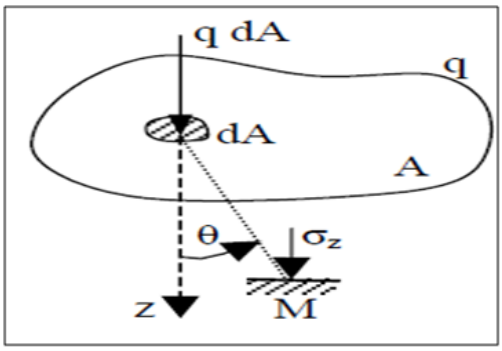



FIGURE I.5 – Charge uniformément repartie

### Cas de surface circulaire

La contrainte verticale en un point situé à la verticale du centre de la surface circulaire uniformément chargée est :

$$\sigma_z = q(1 - \cos^3 \theta_0)$$

Pour le calcul de la contrainte en un point loin de la verticale, il existe des abaques basés sur la méthode des facteurs d'influence

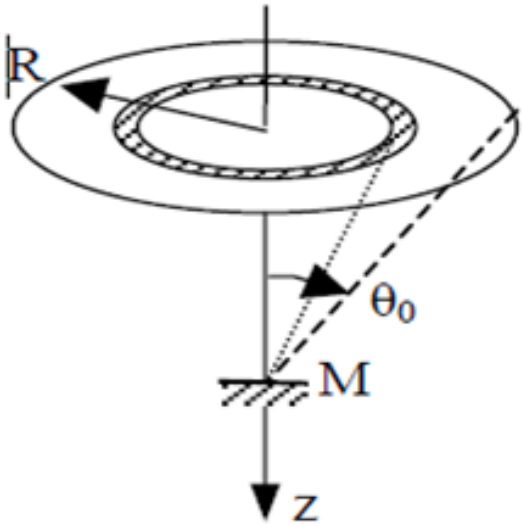



FIGURE I.6 – Charge uniforme repartie Sur une surface circulaire

**Cas de bande rectangulaire**

La contrainte verticale au sein d'un massif à la verticale d'un des sommets est donnée par :

$$\sigma_z = qI(m,n)$$

Où

$$m = \frac{a}{z}$$
$$n = \frac{b}{z}$$

# I.4 Méthodes numérique pour le calcul des contraintes dans le sol

## I.4.1 Intérêt des méthodes numériques

Les méthodes numériques prenant des intérêts considérables. Parmi lesquels :

— Traitement des formes géométriques compliquées

— Traitement des comportements complexes et évolutifs(non linéarité .viscosité Transitoire . . . )

— Couplage des différents phénomènes : mécanique .thermique .magnétismes .hydraulique

— Exploitation directe et rapide des résultats

— Grande capacité d'adaptation aux innovations (modelés .matériaux procèdes)

— Très grand simplicité d'utilisation (c'est aussi un danger Car cela n'encourage Pas trop à réfléchir)

— Néanmoins certains inconvénients peuvent être gérants

— Temps de calcul assez significatif pour les modelés réels

— Grand besoin en ressources informatique (puissance capacité de stockage)

— Résultats non transposables et spécifiques à l'application numériques

— Sensibilités des informations locales pour plusieurs types de problèmes (fissuration . . .

— Effort considérable pour l'interprétation

— Il faut garder àl'esprit que le cout réeld'unemodélisation. n est en réalité ni

— Le cout –machine ni le cout –logiciel. Mais plutôt le cout –ingénieur .évalue par le temps passe pour la définition. l implémentations du modelé la validation l'interprétation et la présentation des résultat

# Chapitre II

# Méthode des éléments finis

## II.1 Introduction

C'est une méthode permettant à l'ingénieur d'effectuer des simulations numérique de phénomène physique avec la moyenne informatique actuelle (puissance du calculateur outils de visualisation) la rende facile de mise en œuvre.

### II.1.1 L'idees de la MEF

Subdiviser (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 forcement De signification physique particulière, mais sont basés sur Des considérations de précision de l'approximation.

### II.1.2 Equation fondamentale de la MEF

La MEF est basée sur la méthode matricielle des déplacements en mécanique des solides et des structures.L'équation fondamentale est :

$$[k]\{U\} = \{F\} \tag{II.1}$$

Avec :

- $[K]$ : matrice de rigidité caractérisant le système $(n \times n)$.
- $\{U\}$ : vecteur des déplacements.
- $DDL$ : Degrés de Libertés $(n)$ inconnus, par exemple, les déplacements et les Rotations des nœuds en mécanique des solides et des structures
- $\{F\}$ : vecteur des sollicitations appliquées $(n)$ connues.

### II.1.3 Etapes logiques du calcul par éléments finis

- maillage.
- le calcul de la matrice de rigidité élémentaire $[K_e]$
- Assemblage
- Modifier le système global en tenant compte des conditions aux limites.
- Résoudre le système $[K]\{U\} = \{F\}$ et obtenir les déplacements $\{U\}$.
- calcul les déformations .
- calcul les contraintes.

### II.1.4 Type de problèmes élément finie

La méthode des éléments finis permet les résolutions de trois types de problèmes Principaux.

### II.1.5 Problèmes d'équilibre stationnaire

Dans ces types de problèmes le comportement est défini en fonction de l'état du système géométrie chargement et des conditions aux limites sous forme d'un système d équation linéaire en fonction des variables nodales on trouve dans cette catégorie l'équilibre statique.

### II.1.6 Problèmes aux valeurs propres

il s'agit des phénomènes de vibration ou d'instabilité d'un état stationnaire .les modes propres de vibration le flambage des structure ou l instabilités des flux laminaire font partie de cette catégorie.

### II.1.7 Problèmes dépendants du temps

Lorsque l'état du système dépend de son histoire ou bien des paramètres de sortie le système devient interdépendante et la résolution directe n'est plus possible ce cas inclus le comportement non linéaire (matériau et géométrie) dynamique non linéaire (amortissement rigidité,) les régimes transitoires (montée en température) et la fissuration des pièces.

## II.2 Discrétisation

### II.2.1 Fonction de forme

le point clé de l'analyse par élément finis de l'interpolation du champ continu $u(x, y, z)$ à partir des valeurs nodales ui ; i étant le numéro du nœud considère il est à rappeler que ce champ u $(x, y, z)$ n 'est qu' une approximation du champ réel $u(x, y, z)$.

L'expression générale de de l'approximation est ainsi donnée par :

$$U(x, y, z) \approx U(x, y, z) = \sum_{noeuds} N(x, y, z).u_i \tag{II.2}$$

Ou $N(x, y, z)$ est la fonction de forme ou (fonction d'interpolation) associée au nœud i et ui est la valeur du déplacement au même nœud les fonctions de forme $N(x, y, z)$ représentent Le poids associe à chacun des nœuds de l'élément elles permettent de prédire l'évolution Du champ à l'intérieur du domaine d'interpolation Pour que l'interpolation soit licite, les fonctions de forme doivent respecter les conditions suivantes :

— Elle doivent être continue sur le domaine, il n'est pas envisageable d'imaginer des sauts entre deux point voisins , une discontinuités signifie que deux points infiniment voisins peuvent avoir des déplacement très diffèrent Or ceci n'est pas conforme à la mécanique du milieu continu (hormis les problèmes de fissuration)

— Pour un jeu unique de valeurs nodales, les fonctions de forme doivent conduite à des valeurs uniques du champ en tout point du domaine.

— Comme dans le cas précèdent un même point physique ne peut pas se déplacer dans Deux directions en même temps (le clonage et interdit)

— Les fonctions de forme doivent satisfaire les valeurs nodales cette condition implique qu'une fonction $N_i(x, y, z)$ doit avoir la valeur 1 au nœud i et 0 a tous les autres nœuds :

$$N_i(x_i, y_i, z_i) = 1 \text{ au } j = i$$
$$0 \text{ aux nœuds } j \neq i$$

Cette condition permet de satisfaire les conditions nodales

$$u(x_i, y_i, z_i) = 1xu_i + \sum_{j \neq 1} 0xu_j \rightarrow u(x_i, y_i, z_i) = u_j$$

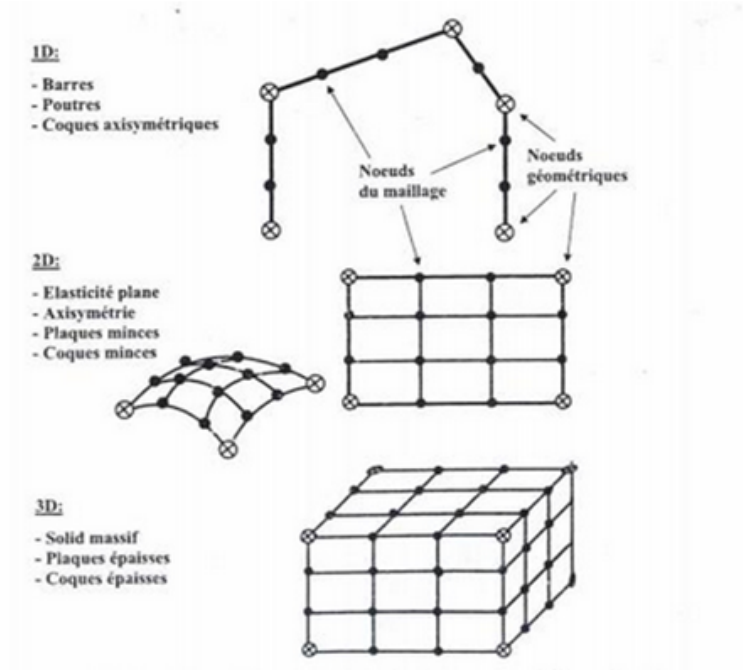



FIGURE II.1 – Diffèrent type élément finis

### II.2.2 Type d'éléments

Il existe de nombreux types d'éléments finis. À choisir en fonction de l'application Des ressources de calcul et de la qualité des résultats. Les figure si après illustrent les types d'élément les plus utilises .le classement se fait en fonction de l'espace et de degré d'interpolations. L'espace d'interpolations correspond au degré de polynôme utilise pour la représentation du champ exact ce degré est directement lié au nombre de nœud dans l'élément. Il peut être linéaire, quadratique, cubique,... En général, Les degrés élevés ne sont pas très courante à cause de la forte augmentation du nombre d'inconnues nodales.

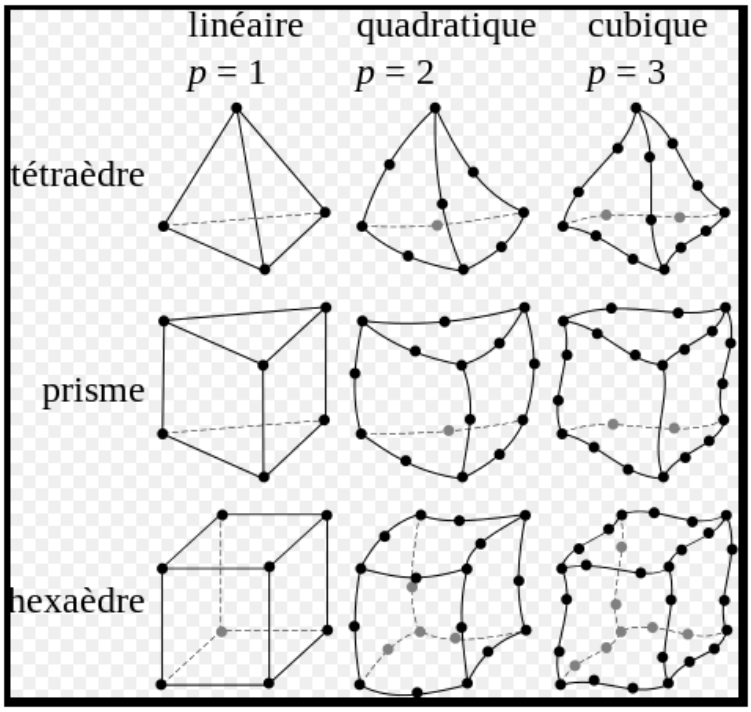



FIGURE II.2 – déférente type d'élément fini 3D

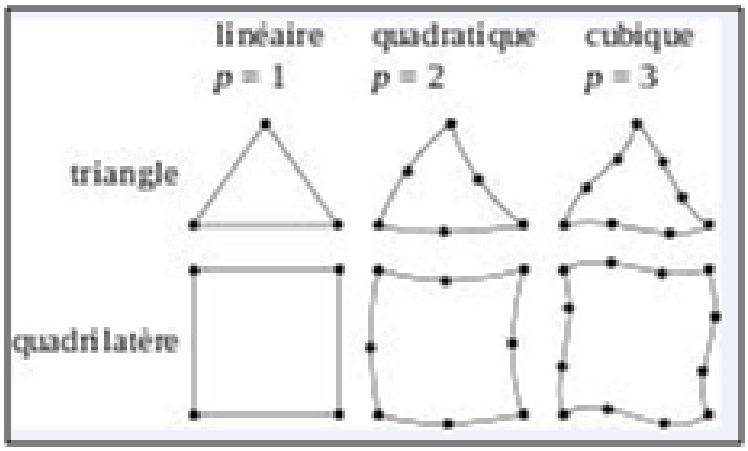



FIGURE II.3 – déférente type d'élément fini 2D

### II.2.3 Fonction de forme D'élément a une dimension

En utilisant les méthodes d interpolation polynomiale, donner les fonctions de forme dès 'éléments a une dimension à deux trois et quatre nœuds .

**Elément a 2 neouds**

Dans le cas des 'élément à deux nœuds l'interpolation doit être linéaire pour assurer l'unicité (Et bien sur la continuité) de la solution car il y a exactement deux coefficients adéterminer par les deux conditions nodales le champ de déplacements s'écrit sous la forme :

$$U(x) = \alpha_0 + \alpha_1 x \tag{II.3}$$

Les deux conditions nodales permettent de déterminer les coefficient $\alpha_0 et \alpha_1$ :

$$U(0) = \alpha_0 + \alpha_1 0 = U_1$$
$$U(L) = \alpha_0 + \alpha_1 L = U_2$$

Ou $u_1$ et $u_2$ sont respectivement les déplacementsaux nœuds 1 et 2
La solution donne les deux coefficients :

— $\alpha_0 = U_1$
— $\alpha_1 = (u_2 - U_1)/L$

Ainsi, on a :

$$U(x) = U_1 + \frac{U_2 - U_1}{L}x \tag{II.4}$$

En regroupant les termes en facteur des valeurs nodales u1 et u2 on obtient :

$$\begin{aligned} U(x) &= (1 - \frac{x}{L})U_1 + (\frac{x}{L})U_2 \\ &= N_1(x)U_1 + N_2(x)U_2 \end{aligned}$$

d ou :

$$N_1(x) = 1 - \tfrac{x}{L} \text{ et } N_2(x)\tfrac{x}{L}$$

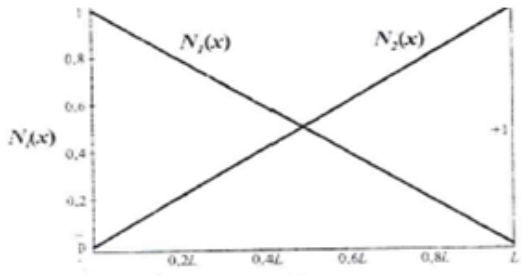



FIGURE II.4 – fonctions de forme de l'élément À 2 nœuds

**Elément a 3 nœuds**

dans ce cas le polynôme doit être quadratique puisqu' il contient trois coefficients

$$U_(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2$$

Les conditions aux nœuds peuvent être écrites sous la forme matricielle suivante

$$\begin{bmatrix} u(0) \\ u(\frac{L}{2}) \\ u(L) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & \frac{L}{2} & \frac{L}{2} \\ 1 & L & L_2 \end{bmatrix} \begin{Bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{Bmatrix} = \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix}$$

Dont la résolutions est :

$$\begin{Bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{3}{L} & \frac{4}{L} & \frac{1}{L} \\ \frac{2}{L} & \frac{4}{L} & \frac{2}{L} \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix}$$

Ou bien :

$$\alpha_0 = u_1$$
$$\alpha_1 = \frac{3}{L}U_1 + \frac{4}{L}U_2 - \frac{1}{L}U_3$$
$$\alpha_2 = \frac{3}{L}U_1 + \frac{4}{L}U_2 - \frac{1}{L}U_3$$

Le champ peut aussi s'écrire sous la forme :

$$u(x) = \sum_{i=1}^{3} N_i(x).U_i = N_1U_1 + N_2U_2 + N_3U_3$$

Avec les fonctions de forme évaluées par :

$$\begin{Bmatrix} N_1(x) \\ N_2(x) \\ N_3(x) \end{Bmatrix} = \begin{bmatrix} 1 & -\frac{3}{L} & \frac{2}{L} \\ 0 & \frac{4}{L} & -\frac{4}{L^2} \\ 0 & -\frac{1}{L} & \frac{2}{L^2} \end{bmatrix} \begin{Bmatrix} 1 \\ x \\ x^2 \end{Bmatrix} = \begin{Bmatrix} 1 - \frac{3}{L}x + \frac{2}{L^2}x \\ \frac{4}{L}x - \frac{3}{L^2}x^2 \\ -\frac{1}{L}x + \frac{3}{L^2}x^2 \end{Bmatrix}$$

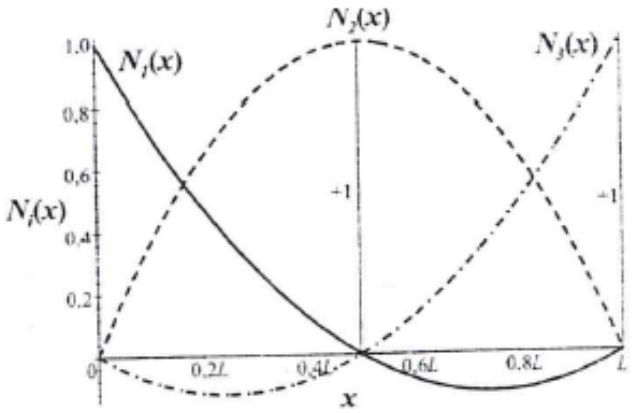



FIGURE II.5 – fonctions de forme de l'élément à 3 nœuds

**Elément à 4 nœuds**

Les procédures est exactement la même que celle dans les cas précédents mais avec un polynôme a quatre coefficient

$$U(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_2 x^3$$

### II.2.4 Les conditions aux nœuds sous la forme matricielle

$$\begin{Bmatrix} U(0) \\ U(\frac{L}{3}) \\ U(\frac{2L}{3}) \\ U(L) \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & \frac{L}{3} & \frac{L}{3} & \frac{L}{3} \\ 1 & \frac{2L}{3} & \frac{2L}{3} & \frac{2L}{3} \\ 1 & L & L & L \end{bmatrix}$$

Si l'élément est défini dans le repère avec extrémités places en $\varepsilon = -1$ et $\varepsilon = +1$ On applique le changement de variable $x = (1+\varepsilon)L/2$ Dans les formules Précédentes des fonctions de forme. Ces dernière sont exprimées dans le nouveau repère par

| FF | 2 nœuds | 3nœuds | 4nœuds |
|---|---|---|---|
| $N1(x)$ | $(1-\varepsilon)/2$ | $-\varepsilon(1-\varepsilon)/2$ | $-(1-\varepsilon)(1-9\varepsilon^2)/16$ |
| $N2(x)$ | $(1+\varepsilon)/2$ | $1-\varepsilon^2$ | $9(1-3\varepsilon)(1-\varepsilon 2)/16$ |
| $N3(x)$ | | $\varepsilon(1+\varepsilon)/2$ | $9(1+3\varepsilon)(1-\varepsilon 2)/16$ |
| $N4(x)$ | | | $-(1+\varepsilon)(1-9\varepsilon 2)/16$ |

TABLE II.1: Fonction de forme des éléments linéiquesà 2 ,3 et 4 nœuds

## II.3 Constructions des fonctions de forme

La construction des fonctions de forme est certainement parmi les étapes les plus importants dans la formation des éléments, la qualité de l'élément en dépend directement surtout son taux de convergence plusieurs méthode existe pour construire les fonctions de forme le choix de la meilleure technique dépend de la nature du comportement désiré Les moyens disponibles et de l'expérience du développeur Les polynômes les plus utilises sont déduits à partir du triangle de pascal. Dans le cas a deux dimension ce triangle est donne par

#### Méthode directe

Cette méthode consiste à choisir un polynôme avec un nombre de coefficient égal à celui des variables nodales pour chacun des champs indépendants, la règle générale est de choisir parmi les termes du triangle de pascal, le polynôme complet de degré minimum ayant un

nombre de coefficients égal au nombre de variables nodales, si ce nombre ne correspond pas à un polynôme complet, il est possible soit de regrouper certains termes soit de les supprimes ,Mais il faut préserver la symétrie de l'élément . À moins que l'élément ne présente pas de symétrie en nombre de nœuds selon les diffèrent axes : dans ce cas, il peut être plus intéressant de privilège certaines directions par rapport à d'autre.

Par exemple, pour un triangle a trois nœuds en élasticité plane les champs de déplacement sont interpoles par (sous l'hypothèse de champs $\mu$ et $v$ indépendants)

$$u(x,y) = \alpha_1 + \alpha_2 x + \alpha_3 y$$
$$V(x,y) = \alpha_4 + \alpha_5 x + \alpha_6 y$$

ou bien :

$$\begin{Bmatrix} u(x,y) \\ v(x,y) \end{Bmatrix} = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & y \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \end{bmatrix}$$

Cette expression s écrit sous une forme compacte par :$\{u\} = [X]\{\alpha\}$ Les conditions nodales permettent de déterminer les valeurs des coefficients Dans notre exemple ces conditions sont données par le système :

$$\begin{bmatrix} u(x_1,y_1) \\ v(x_1,y_1) \\ u(x_2,y_2) \\ v(x_2,y_2) \\ u(x_3,y_3) \\ v(x_3,y_3) \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_1 & y_1 \\ 1 & x_2 & y_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_2 & y_2 \\ 1 & x_3 & y_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_3 & y_3 \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \end{Bmatrix} = \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix}$$

Ce système est note :

$$[A]\{\alpha\} = \{q_e\}$$

Ou la matrice $[A]$ dépend des coordonnées des nœuds, $\{\alpha\}$ est le vecteur des coefficients Et $q_e$ est le vecteur des déplacements nodaux la solution de ce système est donc :

$$\{\alpha\} = [A]^{-1}\{q_e\}$$

Pour simplifier l'écriture. On note :

$$x_{ij} = x_i - x_j$$

$$y_{ij} = y_i - y_j$$

$$c_{ij} = x_i y_j - x_j y_i$$

L'inversion de la matrice $[A]$ donne :

$$[A]^{-1} = \frac{1}{D}\begin{bmatrix} c_{23} & 0 & c_{31} & 0 & c_{12} & 0 \\ y_{23} & 0 & y_{31} & 0 & y_{12} & 0 \\ x_{32} & 0 & x_{13} & 0 & x_{21} & 0 \\ 0 & c_{23} & 0 & c_{31} & 0 & c_{12} \\ 0 & y_{23} & 0 & y_{31} & 0 & y_{12} \\ 0 & x_{32} & 0 & x_{13} & 0 & x_{21} \end{bmatrix}$$

Avec :

$$D = c_{12} + c_{23} + c_{31}$$

Pour obtenir l'expression des fonctions de forme , on introduit ce résultat dans l'équation du déplacement :

$$\{u\} = [X]\{\alpha\} = [X][A]^{-1}\{q_e\}$$

En comparant cette relation avec l'expression générale $\{u\} = [N]\{q_e\}$ On identifie $[N]$par $[N] = [X][A]^{-1}$ Pour le cas de l'élément triangulaire, en posant $d_{ij} = xy_{ij} + yx_{ij}$, on obtient :

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

$$= \frac{1}{D}\begin{bmatrix} c_{23}+d_{23} & 0 & c_{31}+d_{31} & 0 & c_{12}+d_{12} & 0 \\ 0 & c_{23}+d_{23} & 0 & c_{31}+d_{31} & 0 & c_{12}+d_{12} \end{bmatrix}$$

Cela revient a :

$$\begin{Bmatrix} N_1 \\ N_2 \\ N_3 \end{Bmatrix} = \frac{1}{D} \begin{Bmatrix} c_{23} + d_{23} \\ c_{31} + d_{31} \\ c_{12} + d_{12} \end{Bmatrix}$$

Cette méthode est la plus générale pour obtenir les fonctions de forme par interpolation directe du champ de déplacement .néanmoins, elle présente plusieurs inconvénients, parmi lesquels on peut citer :

— Le choix du polynôme est arbitraire, ce qui n'assure pas l'unicité du résultat.

— La lenteur de la méthode est significative à cause de la résolution du système matriciel pour chacun des éléments du maillage.

— La matrice $[A]$ dépend de la géométrie de l'élément et peut être singulière dans plusieurs configurations

### II.3.1 Polynômes de Lagrange

Les polynômes de Lagrange présentent les qualités nécessaires pour la construction des fonctions de formes .Dans le cas unidimensionnel, l'expression du polynôme au nœud i est :

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

Les Propriétés de ces polynômes leur permettent d'interpoler parfaitement le champ de déplacement, ils vérifient les conditions de continuité et d unicités, et possèdent une valeur unitaire au nœud i et nulle nœuds $j \neq i$.

Dans l'exemple de l'élément à 3 nœuds, les polynômes de Lagrange s'écrivent :

$$N_1(x) = L_1(x) = \frac{(x - \frac{L}{2})(x - L)}{(0 - \frac{L}{2})(0 - L)} = (x - \frac{L}{L})(1 - \frac{2x}{L})$$

$$N_2(x) = L_2(x) = \frac{(x - 0)(x - l)}{(\frac{L}{2} - 0)(\frac{L}{2} - L)} = \frac{4x}{L}(1 - \frac{x}{L})$$

$$N_3 = L_3(x) = \frac{(x - 0)(x - \frac{L}{2})}{(L - 0)(L - \frac{L}{2})} = -\frac{x}{L}(1 - \frac{2x}{L})$$

Ce résultat est identique à celui obtenu par la méthode directe il est à noter Que la méthode de Lagrange est bien plus efficace que l'interpolation directe Dans le cas des éléments multi –dimensionnels, il suffit de faire le produit des polynômes de Lagrange dans les diffèrents

directions. Pour cela, une condition essentielle consiste à disposer les nœuds de façon régulière dans chacune des directions orthonormés .dans le cas bidimensionnel, on obtient ce que l'on appelle "Le quadrangle de Lagrange".

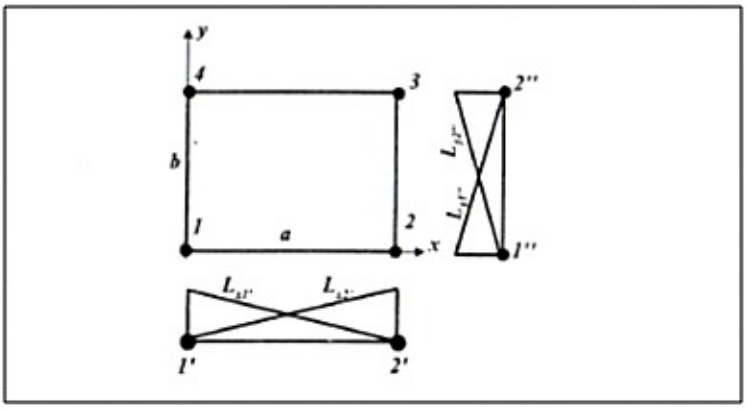



FIGURE II.6 – Fonction de forme d'un élément de Lagrange a quatre nœuds

Chaque rang de nœuds est projeté sur les deux axes $xy$ .cela donne deux nœuds dans chacune des deux directions les fonctions de forme des éléments a une dimension sont facilement établies par les polynômes de Lagrange :

$$L_{x1}(x) = 1 - \frac{x}{\alpha}$$
$$L_{x2}(x) = \frac{x}{\alpha}$$
$$L_{y1}(y) = \frac{y}{b}$$
$$L_{y2}(y) = \frac{y}{b}$$

Pour le quadrangle, on fait la correspondance des coordonnes des nœuds avec celle des nœuds de l'élément linéique :

— Le nœud 1 correspond à 1 et 1 :

$$N_1(x, y) = L_{x1}(x).L_{y1}(y) = (1 - \frac{x}{a})(1 - \frac{y}{b})$$

— Le nœud 2 correspond à 2 et 1 :

$$N_2(x,y) = L_{x2}(x).L_{y1}(y) = \frac{x}{a}(1-\frac{y}{b})$$

— Le nœud 3 correspond à 2 et 2 :

$$N_3(x,y) = L_{x2}(x).L_{y2}(y) = \frac{xy}{ab}$$

— Le nœud 3 correspond à 2 et 2 :

$$N_4(x,y) = L_{x1}(x).L_{y2}(y) = \frac{y}{b}(1-\frac{x}{a})$$

### II.3.2 Polynôme d 'Hermite

les polynôme d Hermite imposent des condition nodales sur le champ et sur ses premières dérivées, ils sont donc utilisée pour les fonction de forme de classe $C^1, C^2$ (Cm indique la continuité de la n$^{\text{eme}}$ dérive) pour construire un polynôme d'Hermite il faut imposer au polynôme choisi des condition sur le champ et ses dérivée a titre d'exemple pour un élément de poutre a deux nœuds la flache et défini par

$$V(x) = N_{1v_v} + N_1\Theta_1 + N_{3v_3} + N_4\Theta_2$$

Ou $N_i$ sont des fonction de forme des flèche et de reptation vi i sont respectivement la flache et la rotation ou nœud i ou extrémité de l'élémentles conditions à respecter sont :

$$v(0) = v_i \Rightarrow N_1 = 1 \text{ et } N_2 = N_3 = N_4 = 0$$

$$v\prime(0) = \Theta_1 \Rightarrow N_1\prime = 1 \text{ et } N_1\prime = N_3\prime = N_4\prime = 0$$

$$v(L) = v_2 \Rightarrow N_3 = 1 \text{ et } N_1 = N_2 = N_4 = 0$$

$$v\prime(L) = v_2 \Rightarrow N_4\prime = 1 \text{ et } N_1\prime = N_2\prime = N_4\prime = 0$$

Avec les autres conditions sur chaque fonction de forme les polynômes doivent être de troisième dégrée Exemple de calcule de polynômed 'Hermite pour un élément de poutre définie entre 0 et L.

Les polynômes choisis troisièmes de degré

$$N_1(x) = \alpha_{0i} + \alpha_{1i}x + \alpha_{2i}x^2 + \alpha_{3i}x^3$$

Pour construire le polynôme $N_i(x)$ on pose le système :

$$\begin{Bmatrix} N_i(0) \\ N_i(0) \\ N_i(L) \\ N_i(L) \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & L & L^2 & L^3 \\ 1 & 1 & 2L & 3L^2 \end{bmatrix} \begin{Bmatrix} \alpha_{0i} \\ \alpha_{1i} \\ \alpha_{2i} \\ \alpha_{3i} \end{Bmatrix} = \begin{Bmatrix} n_{0i} \\ n_{1i} \\ n_{2i} \\ n_{3i} \end{Bmatrix}$$

Avec $n_{ij}$ est la j$^{\text{eme}}$ condition de la fonction $N_i$ dans notre cas $nji$ prend la valeur 1 pour j=i et 0 pour $j \neq i$. l'inversion de ce système donne :

$$\begin{Bmatrix} \alpha_{0i} \\ \alpha_{1i} \\ \alpha_{2i} \\ \alpha_{3i} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\dfrac{3}{L^2} & -\dfrac{2}{L} & \dfrac{3}{L^2} & -\dfrac{1}{L} \\ \dfrac{2}{L^3} & \dfrac{1}{L^2} & -\dfrac{2}{L^3} & \dfrac{1}{L^2} \end{bmatrix} \begin{Bmatrix} n_{0i} \\ n_{1i} \\ n_{2i} \\ n_{3i} \end{Bmatrix}$$

Les coefficients de chacun des polynômes sont tous simplement les colonnes De la matrice ci-dessus .on peut don écrire les fonctions de la forme par substitution des $a_{ij}$ dans l'expression Du polynôme cubique ce qui donne :

$$N_1(x) = 1 - \frac{3x^2}{L^2} + \frac{3x^2}{L^2}$$
$$N_2(x) = x - \frac{2x^2}{L} + \frac{x^2}{L^2}$$
$$N_3(x) = 1 - \frac{3x^2}{L^2} + \frac{3x^2}{L^2}$$
$$N_4(x) = -\frac{x^2}{L} + \frac{x^2}{L^2}$$

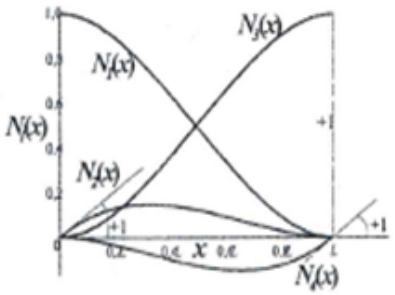



FIGURE II.7 – Elément quadrangulaire à 4 nœuds

### II.3.3 Condition de conformité

Pour assurer une convergence stable des éléments les fonctions de forme doivent satisfaire quatre conditions essentielles, appelées " condition de conformité ≪. Ces conditions offrent la garantie que l'élément est capable de représenter fidèlement le champ réel :

- Assurer la continuité et l'unicité du champ de déplacement dans l'élément Le champ de déplacement approche doit être continu et dérivable en tout point de l'élément .
- assurer la continuité du champ sur la frontière des éléments, ne doit pas se limiter aux nœuds mes sur tout la frontière.
- représenter l'état de déformations constantes dans l'élément.
- représenter le mouvement du corps rigide.

## II.4 Eléments iso paramétriques

Il s'avère très facile de construir des fonctions de forme sur des éléments de forme rectiligne,triangulaire ou cubique que dans la réalité, les structures a étudier sont très différentes de ces formes primitives .il faut donc pouvoir s'approcher de la réalité en disposant de l'élément qui s'adapte a des domaines et des frontières complexes. Dans cette optique, les éléments paramétriques ont été développes afin de permettre de surmonter ces difficultés d'adéquation avec La géométrie réelle.

Le principe de l'élément paramétrique est de définir un élémentde référence Appelé" élément parent " sur lequel on applique une transformation géométrique qui le ramené a la forme de l'élément physique,appelé " élément distordu " (cette terminologie laisse penser que les enfants sont des images distordues de leurs parents ! Pourquoi pas,du moment qu'ils sont plus performent ! )

## II.5 Transformation géométrique

La transformation géométrique permet de définir la forme de l'élément physique à partir d'un certain nombre de points dans l'élément, appelés " points géométrique " .

Les transformations géométriques sont tout simplement des fonctions d interpolation de la géométrie.

Dans le cas plan, on définit un point physique $(x, y)$ en fonction des coordonnes Intrinsèque de l'élément parent $(\varepsilon, \eta)$ et des positions des points de contrôle Géométrique $x_i$ et $y_i$ :

$$x(\varepsilon,\eta) = \sum_{i=1}^{n_g} T_i(\varepsilon,\eta).x_i \qquad\qquad x(\varepsilon,\eta) = \sum_{i=1}^{n_g} T_i(\varepsilon,\eta).y_i$$

Ou $T_i(\varepsilon, \eta)$ sont les fonctions de transformation géométrique ,$n_g$ est le nombre de points géométriques et $x_i, y_i$ sont les coordonnées de ces points.

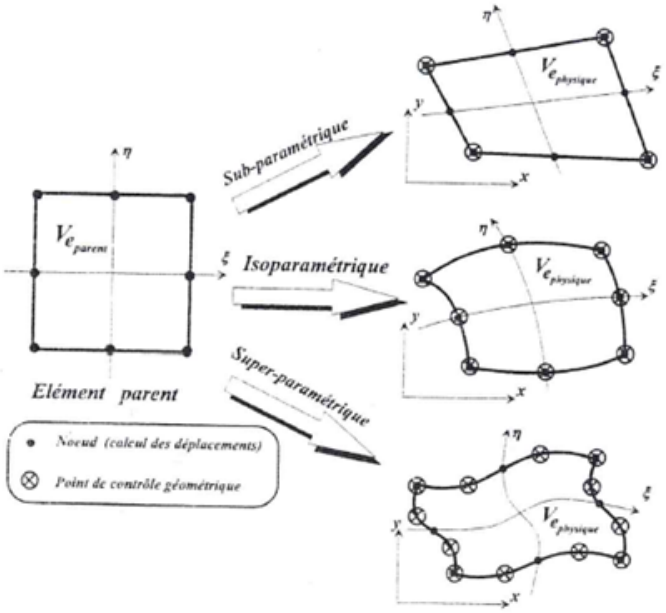



FIGURE II.8 – Diffèrent types d'élément paramétrique

Selon le nombre de points géométrique .il existe trois types d'élément :

— Eléments sub paramétriques : c'est le cas lorsque le nombre de points géométriques est inférieur au nombre de nœuds (ou plus précisément ,Au nombre de degrés de liberté par champ).

— Eléments iso paramétrique : le nombre de points géométriques est égal au nombre de nœuds (plus généralement ,le degré d'interpolation est identique pour la forme

géométrique et les champs de déplacement).

— Eléments super- paramétrique : le nombre de points géométriques est supérieur au nombre de nœuds de l'élément .

### II.5.1 Transformation d'un d élément linéique

Donner la relation géométrique entre les abscisses $x$ et $\varepsilon$,permettant de passer de l'élément parent a l'élément physique ci –dessous.

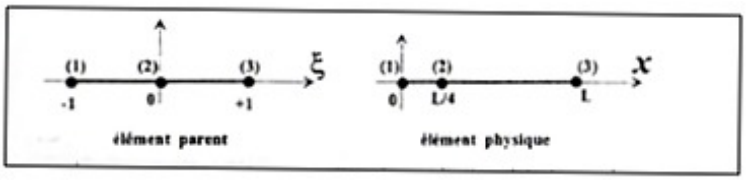



FIGURE II.9 – transformations d'un élément linéique à trois nœuds

Les fonctions d interpolations de l'élément parent sont :

$$T_1 = \frac{1}{2}\varepsilon(1-\varepsilon)$$
$$T_2 = 1-\varepsilon$$
$$T_3 = \frac{1}{2}\varepsilon(1+\varepsilon)$$

La transformation géométrique est donc :

$$x = \sum_{i=1}^{3} T_i x_i = [-\frac{1}{2}\varepsilon(1-\varepsilon)]x_0 + [(1-\varepsilon)]x\frac{L}{4} + [\frac{1}{2}\varepsilon(1+\varepsilon)]xL$$
$$x = \frac{L}{4}(1+\varepsilon)^2$$

On vérifie que les conditions aux limites sont bien respectées :

$$x(-1) = 0$$
$$x(0) = \frac{L}{4}$$
$$x(1) = L$$

## II.6 Matrice de rigidité

La matrice de rigidité d'un élément est initialement définir sur l'élément physique (ou élément distordu) .

$$[K_e] = \int_{V_e physique} [B]^t [D][B] dv$$

Ou

[D ] : Matrice du comportement liant contraint à la déformation.

[B ] : matrice liant déformation au déplacement nodaux et défini comme étant la dérive des fonctions de forme. Ces fonction de forme sont défini sur l'élément parent $N_i(\varepsilon, \eta)$ ou coordonnées $\varepsilon$ et $\eta$ les dérive des fonctions de forme doit être calculer par rapport au coordonnées physique $x$ et $y$

On a

$$\{\frac{\partial N}{\partial \varepsilon}\} = [J]\{\frac{\partial N}{\partial x}\}$$

ou :

[J ] :Matrice jacobéenne formée des dérivée des fonctions $T_i(\varepsilon, \eta)$ Dans l'état plan :

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \varepsilon} & \frac{\partial y}{\partial \varepsilon} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix}$$

Donc :

$$\{\frac{\partial N}{\partial \varepsilon}\} = [J]^{-1}\{\frac{\partial N}{\partial x}\}$$

Pour l'élément iso paramétrique, les points géométriques sont les nœuds servant à la discrétisation du champ, Dans ce cas, les propriétés des fonctions de forme sont parfaitement adaptées à la définition de la transformation géométrique, on prend donc $T_i = N_i$ ,Ainsi, la matrice $[B]$ est obtenue en fonction des coordonnées intrinsèque (de l'élément parent) et des coordonnées des nœuds dans l'espace physique.

Pour l'intégration de la matrice de rigide, il reste à définir la distorsion des éléments de volume .l'intégration doit se faire sur le volume de l'élément physique .tandis que l'intégrant est fonction du repère de l'élément parent. Il faudra donc effectuer l'intégration sur l'élément parent.

Donc :

$$[K_e] = \int_{V_e physique} [B]^t[D][B]D_v = \int_{V_e parent} [B]^t[D][B][J]dv_{parent}$$

## II.6.1 Formulation de la matrice b

**Elément finie choisi**

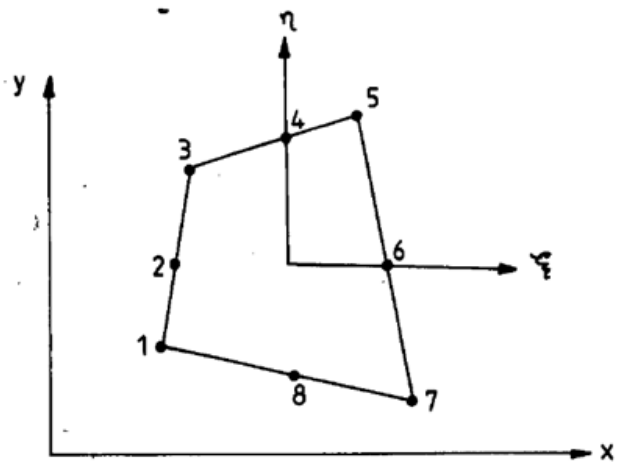



FIGURE II.10 – Elément quadrangulaire à 8 NŒUDS

**Calcules de fonction de forme pour élément quadrangulaire à 8 NŒUDS**

Cet élément est quadratique chacun de ces segments comportant trois nœuds Ceux-ci permettent une approximation parabolique sur chacun de ses cotes Néanmoins , la fonction d'approximation aurait pu être totalement parabolique sans le nœud :

$$u(\varepsilon,\eta) = \sum_{i=1}^{8} N_i.U_i$$

$$u(\varepsilon,\eta) = a_0 + a_1\varepsilon + a_2\eta + a_3\varepsilon\eta + a_4\varepsilon^2 + a_5\eta^2 + a_6\varepsilon\eta^2 + a_7\eta\varepsilon^2$$

$$u(\varepsilon,\eta) = \{a\}^t.\{X\}$$

Avec :

$$\{a\}^T = \begin{Bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \end{Bmatrix}$$

$$\{X\}^T = \begin{Bmatrix} 1 & \varepsilon & \eta & \varepsilon\eta & \varepsilon^2 & \eta^2 & \varepsilon\eta^2 & \eta\varepsilon^2 \end{Bmatrix}$$

$$\begin{bmatrix} U(1,1) \\ U(0,1) \\ U(-1,1) \\ U(-1,0) \\ U(-1,-1) \\ U(0,-1) \\ U(1,0) \end{bmatrix} = [R]\{a\}$$

$$[R] = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

On déduit pour les fonctions de forme de l'élément :

$$
\begin{aligned}
&\{N\} = \{X\}^t[R] \\
&N_1 = -\frac{1}{4}(1+\varepsilon)(1+\eta)(1-\varepsilon-\eta) \\
&N_2 = -1\frac{1}{2}(1-\varepsilon)(1-\eta^2) \\
&N_3 = -\frac{1}{4}(1-\varepsilon)(1+\eta)(-\varepsilon+\eta-1) \\
&N_4 = -\frac{1}{2}(1-\varepsilon^2)(1+\eta) \\
&N_5 = -\frac{1}{4}(1+\varepsilon)(1+\eta)(\varepsilon+\eta-1) \\
&N_6 = -\frac{1}{2}(1+\varepsilon)(1-\eta^2) \\
&N_7 = -\frac{1}{4}(1+\varepsilon)(1-\eta)(\varepsilon-\eta-1) \\
&N_8 = -\frac{1}{2}(1-\varepsilon^2)(1-\eta)
\end{aligned}
$$

$$
[B] = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial x} & \frac{\partial N_i}{\partial y} \end{bmatrix} i = 1
$$

$$
O_r[B] = \begin{bmatrix} \frac{\partial n_i}{\partial r} & 0 \\ \frac{1}{r} & 0 \\ 0 & \frac{\partial N_i}{\partial Z} \\ \frac{\partial N_i}{\partial Z} & \frac{\partial N_i}{\partial r} \end{bmatrix} i = 1
$$

Dans le cas des repères cylindriques choisis pour un problème axisymétrique

### II.6.2 MatriceD

la matrice D défini par la relation contrainte déformation (l'état plan de contrainte).

$$
\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{xx} \\ 2\varepsilon_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} U_x \\ U_y \end{bmatrix} [\varepsilon] = [A][U]
$$

$$[A] = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial x} \end{bmatrix}$$

### II.6.3 Forces élémentaires

Pour calculer les force élémentaires, il suffit d'effectuer l'intégration sur l'élément parant en faisant intervenir le jacobien pour l'intégrale de volume correspondantaux forces volumiques et aux contraint résiduelle

$$\{F_e^v\} = \int_{Veparent} [N]^t \{f^v\} |J| dv$$
$$\{F_e^o\} = \int_{Veparent} [N]^t \{\sigma^o\} |J| dv$$

Pour les forces surfacique l'élément d'intégration Ds est écrits dans le cas 2D par :

$$ds = \sqrt{dx^2 + dy^2} = \sqrt{((\frac{\sigma x}{\sigma \varepsilon} d\varepsilon) + (\frac{\sigma x}{\sigma \eta} d\eta))^2 ((\frac{\sigma y}{\sigma \varepsilon} d\varepsilon) + (\frac{\sigma y}{\sigma \eta} d\eta))^2}$$

Sur la base de ces expressions on désignera par $|J_s|$ le terme en facteur de la coordonnée intrinsèque considérée Ce qui permet d'écrire les forces surfacique sous la forme

$$\{F_e^s\} = \int_{Separent} [N]^t \{f^s\} |J_s| ds$$

#### Integration Numerique

L'intégration pour calculer la matrice de rigidité est impossible pour cela le choix d'une méthode d'intégration numérique et obligatoire

#### Méthodes de trapèzes

Cette méthode basée sur la division le volume de l'élément en un certain nombre de régions. L'intégration sur chaque de ces régions est approches par le volume de la région multiplie par la valeur moyenne de l'intégrationsur tout l'élément est obtenus par la somme des surfaces des trapèzes .à cause de la faible précision cette méthode n'est pas utilisé en méthode des elements finis

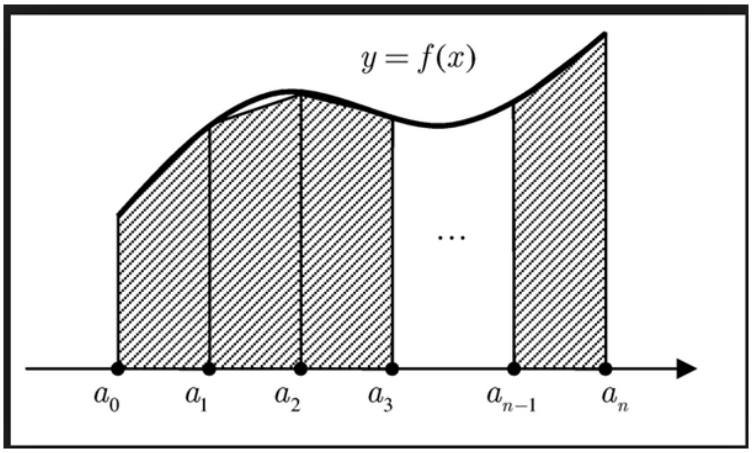



FIGURE II.11 – méthode de trapèzes

**Méthodes de Simpson**

Ces méthodes sont basées sur une approximation parabolique. l intégrations d'une Fonction est obtenue par :

$$\int_a^b f(x)dx = \frac{h}{3}[f(a) + 4f(a+h) + 2f(a+2h) + 4f(a+3h) + 2f(a+4h) + \ldots$$
$$+2f(b-2h) + 4f(b-h) + f(b)]$$

Avec $h = (b-h)/n$ et $n$ est le nombre d'intervalles qui doit être pair. Cette méthode est nettement meilleure que la méthodeprécédente, mais elle reste inefficace pour calculer les matrices des éléments.

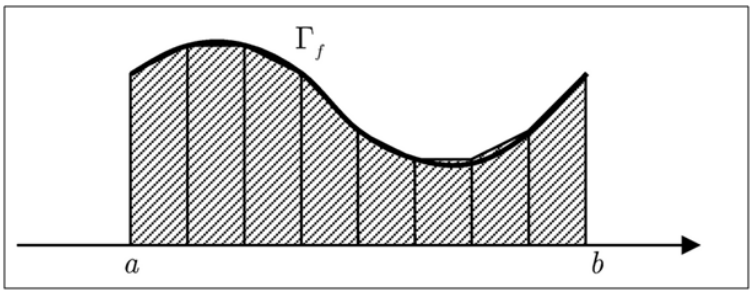



FIGURE II.12 – Méthodes de Simpson

**Méthode de gauss** C'est une méthode efficace basée sur les polynômes orthogonaux de Legendre. Dans cette méthode les positions des points d'intégration sont définies pour optimiser l'approximation polynomiale. Pour expliquer la philosophie de la méthode, on prend l'exemple de l'intégration d'un polynôme du troisième degré :

$$\int_a^b (a_0 + a_1\varepsilon + a_2\varepsilon + a_3\varepsilon)^3 d\varepsilon = 2a_0 + \frac{2}{3}a_2$$

Si on considère deux points d'intégration de poids identique ($w_1 = w_2$) et symétriques par rapport à l'origine ($\varepsilon 1 = -\varepsilon 2$) la somme sur ces deux points s écrit :

$$\sum_{i=1}^{2} w_1(a_0 + a_1\varepsilon + a_2\varepsilon^2 + a_3\varepsilon^3) = 2w_1(a_0 + a_2\varepsilon^2)$$

L'intégration numérique sera exacte avec seulement deux points et :

$$2w_1(a_0 + a_2\varepsilon^2) = 2a_0 + \frac{2}{3}a_2$$

La seule solution est donc : $w_1 = w_2 = 1$ et $\varepsilon 1 = -\varepsilon_2 = \frac{1}{\sqrt{3}}$ . Les valeurs de $\varepsilon_i$ et $w_i$ Sont Fournies dans les tableaux en fonction du nombre de Gauss .donc l'intégration devient.

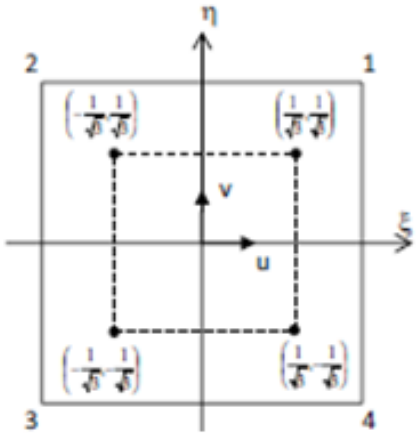



FIGURE II.13 – Méthodes de Gauss

| $N$ | $x_i, y_i$ | $w_i, w_J$ |
|---|---|---|
| 1 | 0 | 2 |
| 2 | $-\frac{1}{\sqrt{3}} \quad \frac{1}{\sqrt{3}}$ | 1 1 |
| 3 | $-\frac{\sqrt{5}}{5} 0 -\frac{\sqrt{5}}{5}$ | $\frac{5}{9} \frac{8}{9} \frac{5}{9}$ |

TABLE II.2: Integration dans le sol

$$\int_1^1 f(\varepsilon) d\varepsilon \sum_{i=1}^{nG} w_i f(\varepsilon_i)$$

Pour les intégrales de surface une double sommation est utilisée :

$$\int_{-1}^{+1} \int_{-1}^{+1} f(\varepsilon.\eta) d\varepsilon d\eta = \sum_{i=1}^{nG\varepsilon} \sum_{j=1}^{nG\eta} w_i w_i f(\varepsilon_i, f\eta_j)$$

### II.6.4 Assemblage des matrice de rigidité

Dans la pratique un structure et modéliser par un certain nombre d'éléments l'énergie potentiel De la structure et donne par la somme des énergies de ses éléments :

$$\prod = \sum_{i=1}^{e} \prod$$

De même le variation virtuelle de l'énergie structurale vérifier le principe de stationnarité :

$$\delta \prod = \sum_{e}^{i=1} \delta \prod_{e}$$

On définit un vecteur représentant les déplacementsdes nœuds de la structure $\{q\}$ le vecteur des déplacementsdes nœuds d'un élément $\{q_e\}$ est obtenue a l'aide d'une matrice de sélection $[s_e]$.un terme de cette matrice est Soit 0 soit 1 (pour les degrés de liberté sélectionnées)

$$\{q_e\} = [s_e]\{q\}$$

cette sélection donne le vecteur $\{q_e\}$ dans le repère de la structure (repère de $\{q_e\}$ ). Il faudra donc passer au repère de l'élément en utilisant la matrice de passage $\{T_e\}$

$$\{q_e\}_{local} = \{T_e\}[s_e]\{q\}$$

Il reste maintenant a vérifier la stationnarité de l'énergie potentielle de la structure :

$$\delta\prod = \sum_e \{\delta q\}^t [s_e]^t [T_e]^t (\int [B]^t [C][B] dv)[T_e][S_e]\{q\} - \{\delta q\}^t [s_e]^t [T_e]^t \int_{v_e} [B^t][C]\{\varepsilon^0\} dv$$
$$- \{\delta q\}^t [S_e]^t [T_e]^t \int_{v_e} [N]^t |f^v| dv - \{\delta q\}^t [S_e]^t [T_e]^t \int_{s_e} [N]^t |f^s| ds$$

Cette expression permet de déduire les quantités élémentaires dans le repere structural :

$$[K_e]_{global} = [T_e]^t [K_e]_{local} [T_e]$$
$$\{F_e^v\}_{global} = [T_e]^t \{F_e^v\}_{local}$$
$$\{F_e^s\}_{global} = [T_e]^t \{F_e^s\}_{local}$$
$$\{F_e^0\}_{global} = [T_e]^t \{F_e^0\}$$

La stationnarité de l'énergie potentielle conduit au système d'équilibre structural : $[k]$ avec la matrice de rigidité de la structure $\{q\}$ est le vecteur des déplacement nodaux et le vecteur Des forces nodales .l'énergie interne et ainsi donne par :

$$\prod = \frac{1}{2}\{q^t\}[k]\{q\}$$

# Chapitre III

# Les problemes axisymetrique

## III.1 Introduction

Pour calculer les contraint dans le sol par la méthode des éléments finis, le sol est considère comme un milieu continu (semi infini) qui peux être assimilé comme corps axisymétrique De plus si on peut avoir le problème en 2D c'est le cas idéal, réellement c'est applicable. Si le chargement est axisymétrique alors le problème tridimensionnel se ramène a un problème bidimensionnel, le traitement dans ce cas est similaire a ceux des contraintes planes et de des déformations planes, Donc avec cette méthode on peut ramener facilement le problème 3D a un problème 2D. Donc si on utilise ces deux approches (axisymetrie de la structure et le traitement du problème en 2D) on réduit énormément la capacité du stockage et de traitement dans la machine, donc on peut utiliser facilement les machines ultra sophistiquées disponibles dans le marché.

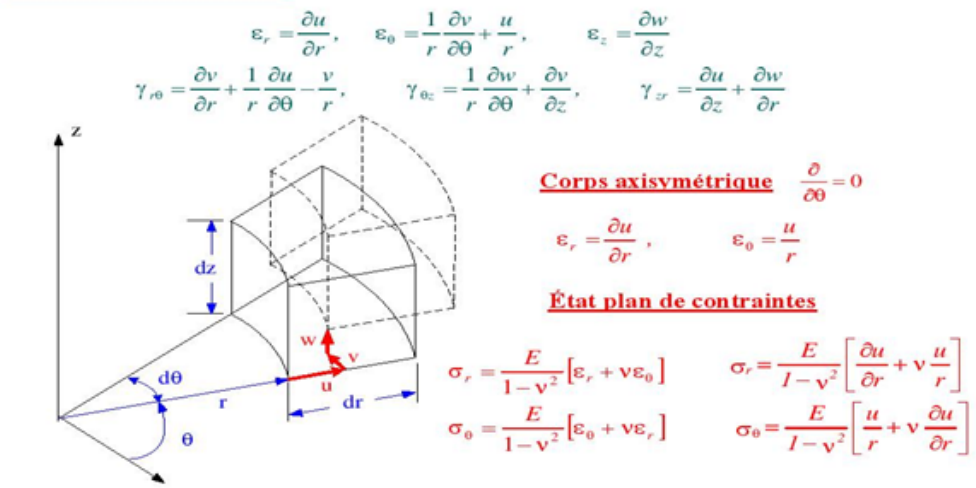



FIGURE III.1 – Déformations-déplacements-contraints

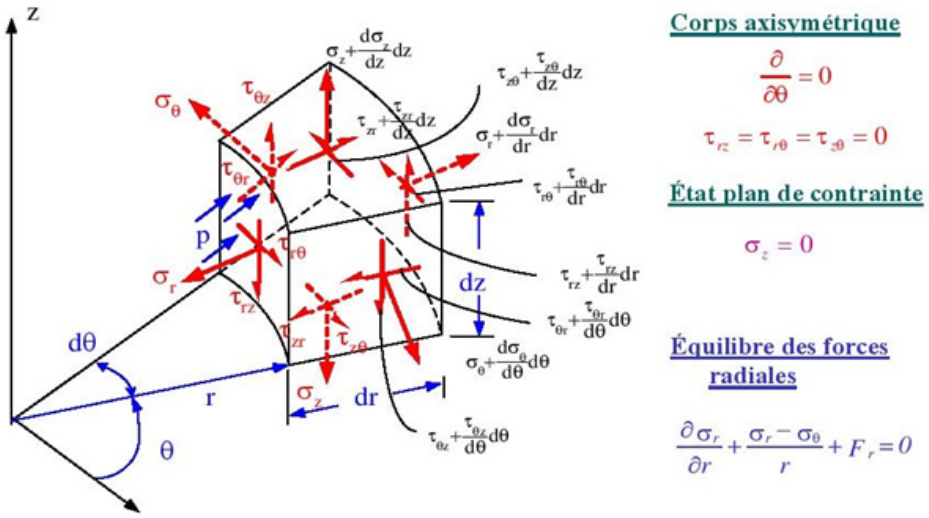



FIGURE III.2 – Corps axisymétrique

## III.2 Elément finie choisi

Dans notre cas on a choisi les éléments quadrangulaire a 8 nœuds qui sont utile dans note exemple

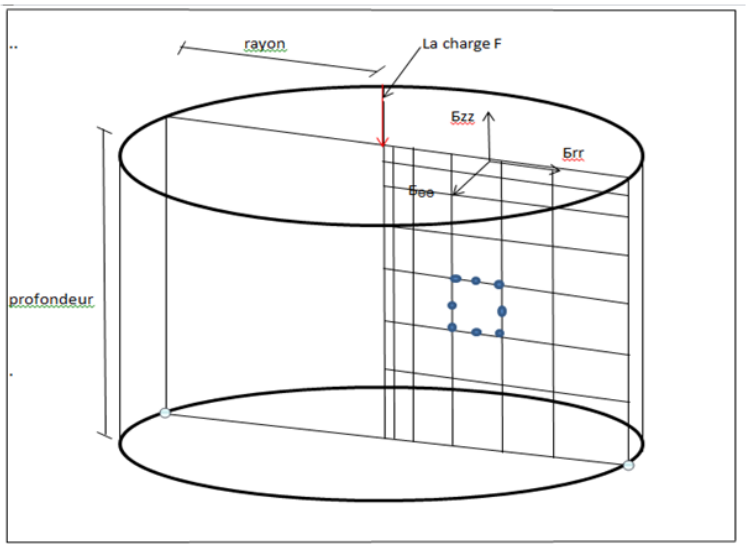



FIGURE III.3 – Elément choisi

### III.2.1 Les éléments quadrangulaires à 8 nœuds

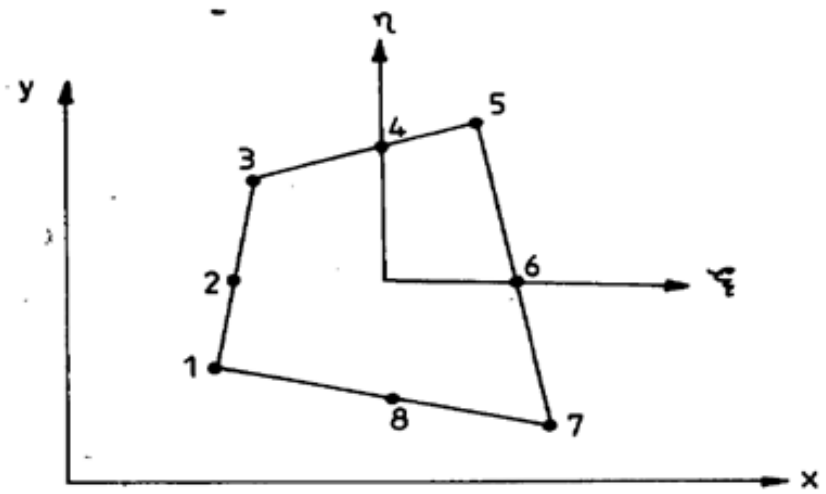



FIGURE III.4 – Eléments quadrangulaires à 8 nœuds

On A choisir les éléments quadrangulaires à 8 nœudsà savoir 4 nœuds aux sommets et 4 nœuds aux milieux, on définit un élément iso paramétrique de référence dans un repère local E et y Élément référence est transformé par un élément réel par une transformation géométrique qui défini x de chaque point de l'élément réel a partir de E de l'élément référence. On peut exprimer les fonctions d'interpolations et leurs dérivées partiels dans le tableau suivant :

| | N | $\frac{\partial N}{\partial \varepsilon}$ | $\frac{\partial N}{\partial \eta}$ |
|---|---|---|---|
| 1 | $\frac{-(1-\varepsilon)(1-\eta)(1+\varepsilon+\eta)}{4n}$ | $\frac{(1-\eta)(2\varepsilon+\eta)}{4}$ | $\frac{-(1-\varepsilon)(\varepsilon+2\eta)}{4}$ |
| 2 | $\frac{(1-\varepsilon^2)(1-\eta)}{2}$ | $-1(1-\eta)\varepsilon$ | $-\frac{(1-\varepsilon^2)}{2}$ |
| 3 | $\frac{-(1-\varepsilon)(1-\eta)(1-\varepsilon+\eta)}{4}$ | $\frac{(1-\eta)(2\varepsilon-\eta)}{4}$ | $-\frac{(1-\varepsilon^2)}{2}$ |
| 4 | $\frac{(1-\eta^2)(1+\varepsilon)}{2}$ | $\frac{(1-\eta^2)}{4}$ | $\frac{-(1-\varepsilon)(\varepsilon-2\eta)}{2}$ |
| 5 | $\frac{-(1+\varepsilon)(1-\eta)(1-\varepsilon-\eta)}{4}$ | $\frac{(1-\eta)(2\varepsilon-n)}{2}$ | $\frac{(1+\varepsilon)(\varepsilon-2\eta)}{4}$ |
| 6 | $\frac{(1-\varepsilon^2)(1+\eta)}{2}$ | $-1(1+\eta)\varepsilon$ | $\frac{(1-\varepsilon^2)}{2}$ |
| 7 | $\frac{-(1-\varepsilon)(1+\eta)(1+\varepsilon-\eta)}{4}$ | $\frac{(1+\eta)(2\varepsilon-\eta)}{4}$ | $\frac{-(1-\varepsilon)(\varepsilon-2\eta)}{4}$ |
| 8 | $\frac{(1-\eta^2)(1-\varepsilon)}{2}$ | $-\frac{(1-\eta^2)}{2}$ | $-1(1-\varepsilon)\eta$ |

TABLE III.1: Fonctions d'interpolations et leurs dérivées partielles

## III.3 Matrice de rigidité

La matrice de rigidité élémentaire est donnée par :

$$[k]^e = \iint^e 2.\pi[B]^T[D][B]rdrdz$$

Ou B est la matrice de forme :

$$[B] = \begin{bmatrix} \frac{\partial N_i}{\partial r} & 0 \\ 0 & \frac{\partial N_i}{\partial z} \\ \frac{N_i}{r} & 0 \\ 0 & \frac{N_i}{r} \end{bmatrix} (1 \leq i \leq 8)$$

et :

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

Pour exprimer les dérivés de l'élément réel en fonction de l'élément référence :

$$\begin{bmatrix} \frac{\partial N_i}{\partial r} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} = [j]^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial \varepsilon} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix}$$

Avec $[J]$ est la matrice jacobienne de la transformation :

$$[J] = \begin{bmatrix} \frac{\partial N_1}{\partial \varepsilon} & \frac{\partial N_2}{\partial \varepsilon} & \cdots & \frac{\partial N_8}{\partial \varepsilon} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \cdots & \frac{\partial N_8}{\partial \eta} \end{bmatrix}$$

L'expression de la matrice de rigidité devient dans le repère local :

$$[K]^e = \int_{-1}^{1}\int_{-1}^{1} 2\pi \sum_{i=1}^{8} r_i^{(e)} N_i(\varepsilon,\eta) det(j)([B]^T[D][B]) d\varepsilon d\eta$$

Soit :

$$[K]^e = \int_{-1}^{1}\int_{-1}^{1} G(\varepsilon,\eta) d\varepsilon d\eta$$

Le calcul de la matrice de rigidité élémentaire $[K]^e$ se fait numériquement par l'utilisation de l'intégration de Gauss donc le calcul mathématique de chaque terme de la matrice de rigidité sera remplacé par une sommation aux niveaux des points de Gauss, dans notre cas on utilise seulement deux points.

L'écriture de la matrice de rigidité élémentaire sera remplacée par :

$$[K]^e = \sum_{i=1}^{n}\sum_{j=1}^{m} w_i w_j G(\varepsilon,\eta)$$

Avec : n et m sont respectivement le nombre de points de Gauss dans la direction $\varepsilon$ et $\eta$ et $H_i$ et $H_j$ sont les coefficients correspondants.

Après le calcul de la matrice de rigidité élémentaire aux niveaux de chaque élément, le programme assemble tous ces matrices élémentaires dans une matrice globale $[K]$.On a généralement un système d'équation $[K].\{u\} = \{F\}$ Ou F est le vecteur représentant toutes les sollicitations extérieures. Les déformations Relation contrainte déformations dans l'état plane .

### III.3.1 Condition ou limite

Les conditions aux limites sont indispensables pour rendre la matrice de rigidité inversible. En général des conditions sont imposées sur certain déplacement de la structure. Les valeurs imposées peuvent être nulles, fixées ou relative a d'autres points structuraux

# III.4 Les deplacement

## III.4.1 Cas 1-deplacement nul $u_i$

Il surfait de remplacer la ligne et la colonne i par "0" sauf sur la diagonale, on met "1" le second membre est également remplacer par zéro .cette opérations remplacer par zéro cette opérations remplace l'équations du terme $u_i$ par l'équations :$u_i = 0$.

$$\begin{bmatrix} & & 0 & & \\ 0 & \dots & 1 & \dots & 0 \\ & & 0 & & \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_i \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ 0 \\ \vdots \\ f_n \end{bmatrix}$$

## III.4.2 Cas 2- déplacement imposer non nul $u_i = \overline{u_i} \neq 0$

Le term diagonal $k_{ii}$ est multiplie par un nombre important ($10^20$par exemple)t le terme second membre est remailer par $10^20 k_{ii}\overline{u_i}$ le system devient :

$$\begin{bmatrix} & & k_{1i} & & \\ k_{i1} & \dots & 10^20 k_{ii} & \dots & k_{in} \\ & & k_{ni} & & \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_i \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ 10^20 k_{ii} \\ \vdots \\ f_n \end{bmatrix}$$

avec n la taille de la matrice (degré de liberté de la structure).cette technique permet d'écrire l'équation du degré de liberté imposer sous la forme :

$$u_1 = \frac{10^{20} k_{ii}\overline{u_i}(k_{i1}u_1 + k_{i2}u_2 + \cdots + k_{i(i-1)}u_{i-1} + k_{i(i+1)}u_{i+1} + \cdots + k_{in}u_{in})}{10^{20} k_{ii}}$$
$$\approx \frac{10^{20} k_{ii}\overline{u_i}}{10^{20} k_{ii}} = \overline{u_1}$$

Par rapport au cas 1 ,cette formulation permet de conserver l'interdépendance entre le déplacement imposer ($\overline{u_i}$)et les autre déplacement du system $u_j \neq i$

## III.5 Les contraintes

dans la des problèmes axisymétriques le tenseur des contraintes et calculer par la relation suivant :

$$\begin{bmatrix} \sigma_t \\ \sigma_\phi \\ \sigma_z \\ \tau_{zr} \end{bmatrix} = \frac{(1-v)E}{(1+v)(1-2v)} \begin{bmatrix} 1 & f & f & 0 \\ f & 1 & f & 0 \\ f & f & 1 & 0 \\ 0 & 0 & 0 & g \end{bmatrix} \begin{bmatrix} \varepsilon_t \\ \varepsilon_\phi \\ \varepsilon_z \\ \gamma_{zr} \end{bmatrix}$$

D ou :

$$f = \frac{v}{1-v}$$

and :

$$f = \frac{1-2v}{2(1-v)}$$

# Chapitre IV

# Présentation de Language Visuel Basic 6

## IV.1 Les principaux concepts de Visual Basic

### IV.1.1 Présentation générale

Visual Basic est un langage de programmation « orienté objet » de Microsoft qui permet deprogrammer des applications indépendantes sous l'environnement Windows. Il est intégré dans tous les logiciels de Bureautique de Microsoft (Word, Excel, Access) sous le nom de : VBA (Visual Basic Application). Visual Basic est un langage interprétéPour comprendre le processus de développement d'une application, il est utile d'assimiler certains concepts sur lesquels est fondé Visual Basic. Comme il s'agit d'un langage de développement Windows, il convient également de s'être familiarisé avec l'environnement Windows. Le fonctionnement de Windows s'articule autour de trois concepts essentiels.

— les fenêtres .

— les événements .

— les messages.

Considérons qu'une fenêtre est une zone rectangulaire dotée de ses propres limites. Nous connaissons tous une fenêtre de document dans un programme de traitement de texte ou une fenêtre de boite de dialogue quelconque. S'il s'agit là des exemples les plus courants,

il existe bien d'autres types de fenêtres. Un bouton de commande, les icônes, les zones de texte, les boutons d'options, les barres de menus constituent tous des fenêtres.

Windows gèrent ces nombreuses fenêtres en affectant à chacune d'elles un numéro d'identification unique (« hWnd»). Le système surveille en permanence chacune de ces fenêtres de façon à déceler le moindre événement.

Les événement peuvent être engendrés par des actions de l'utilisateur, par un contrôle programmé, voire par des actions d'une autre fenêtre. Chaque fois qu'un événement survient, un message est envoyé au système d'exploitation, qui traite le message et le diffuse aux fenêtres concernées. La plupart de ces messages de bas niveau sont gérés par Visual Basic tandis que d'autres sont mis à votre disposition sous forme de procédures événementiels.

## IV.2 Le modèle événementiel

Dans les applications traditionnelles, dites « procédurales », les instructions commencent à la première ligne de code et se déroulent suivant un chemin défini dans l'application, appelant les procédures au fur et à mesure des besoins. Dans une application « événementielle », le code ne suit pas un chemin prédéterminé. Différentes sections du code sont exécutées en fonction des événements qui se produisent. Vous ne pouvez prévoir l'ordre des événements, en revanche vous devez prévoir les événements qui peuvent se produire et votre code doit être en mesure de les traiter.

### IV.2.1 La notion d'objet

Un objet représente un élément d'une application. Une feuille de calcul, une cellule, ungraphique pour Excel, un formulaire, une table ou un état pour Access sont des objets. Dans uncode Visual Basic, vous devez identifier un objet avant de pouvoir appliquer l'une des méthodesde l'objet ou modifier la valeur de l'une de ses propriétés.

Une collection est un objet contenant plusieurs autres objets, généralement, mais pas toujours, du même type. Dans Microsoft Excel, par exemple, l'objet Workbooks contient tous les objetsWorkbook. Dans Visual Basic, la collection Forms contient tous les objets Form d'uneapplication.Les éléments d'une collection peuvent être identifiés par numéro ou

par nom.

Vous pouvez également manipuler toute une collection d'objets si les objets partagent des méthodes communes.

### IV.2.2 Les notions de méthode et de propriété

Une méthode est une action qu'un objet peut exécuter. Par exemple, Add est une méthode de l'objet Combo Box qui ajoute une nouvelle entrée à une liste modifiable.Une propriété est un attribut d'un objet définissant l'une des caractéristiques de l'objet telle quela taille, la couleur ou la position à l'écran, ou un aspect de son comportement, par exemple s'il est activé ou visible. Pour changer les caractéristiques d'un objet il faut changer les valeurs de ses propriétés.Pour définir la valeur d'une propriété, faites suivre la référence d'un objet d'un point, du nom de la propriété, d'un signe égal (=) et de la nouvelle valeur de propriété. Certaines propriétés ne peuvent pas être définies. La rubrique d'aide de chaque propriété indiquesi vous pouvez la définir (lecture-écriture), seulement la lire (lecture seule) ou seulement y écrire(écriture seule). Vous pouvez extraire des informations sur un objet en renvoyant la valeur del'une de ses propriétés.

## IV.3 Prise en main de l'environnement

### IV.3.1 Démarrage de Visual Basic

Au démarrage de Visual Basic, on obtient la fenêtre suivante :

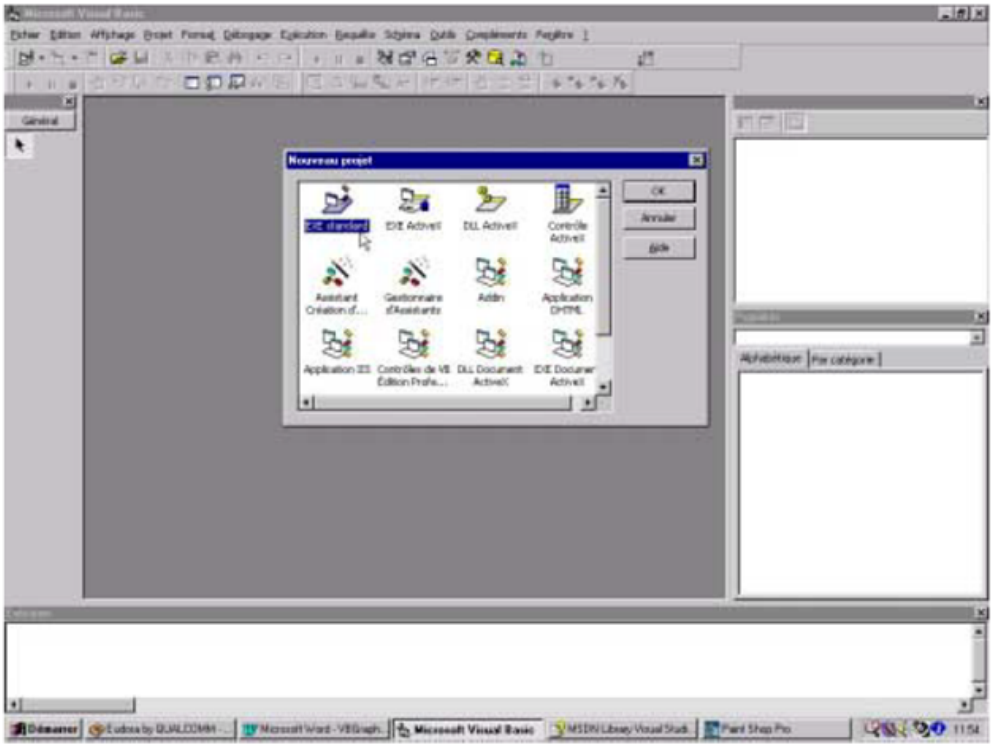


FIGURE IV.1 – Fenetre Démarrage de Visual Basic

Le choix le plus couramment utilisé lors de la création d'un nouveau projet est l'application exécutable standard (EXE standard). Dans ce document, nous n'aborderons que ce type de projet. Si vous désirez ouvrir un projet existant, cliquez sur le bouton « annuler » de la fenêtre Nouveau « projet » puis cliquez sur « Ouvrir un projet » dans le menu « Fichier ».

### IV.3.2 Les éléments de l'environnement

L'environnement comporte sept éléments principaux : la fenêtre code, la fenêtre de conception de feuille, l'explorateur de projet, la fenêtre des propriétés, la fenêtre d'exécution et la boite à outils.

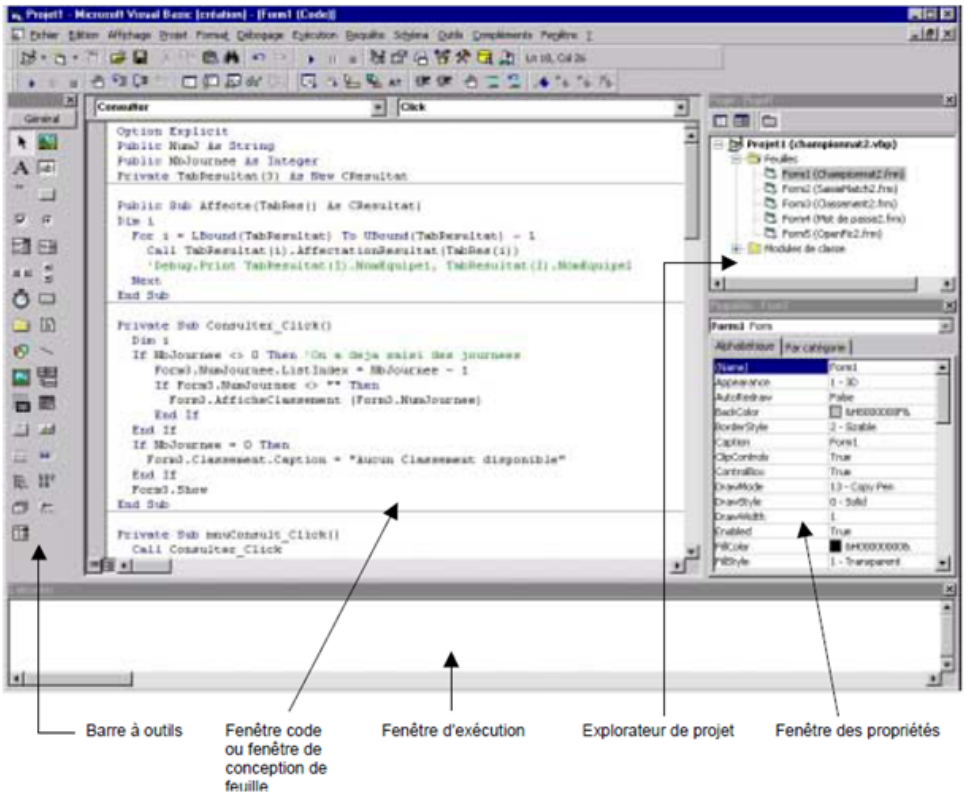



FIGURE IV.2 – Les éléments de l'environnement

### Fenêtre Code

Elle fait office d'éditeur pour la saisie du code de l'application. Une fenêtre Code distincte est créée pour chaque feuille ou module de code de votre application.

### Fenêtre de conception de feuille

Elle fait office de fenêtre personnalisable pour la création de l'interface de votre application. Vous ajoutez des contrôles, des graphismes et des images à une feuille de façon à ce qu'elle prenne l'apparence souhaitée. Chaque feuille de votre application possède sa propre fenêtre de conception de feuille.

### Fenêtre Explorateur de projets

Cette fenêtre énumère les feuilles et les modules contenus dans votre projet en cours. Un projet est un ensemble de fichiers à partir desquels vous créez une application.

### Fenêtre Propriétés

Cette fenêtre énumère les paramètres des propriétés de la feuille ou du contrôle sélectionné. Une propriété définit une caractéristique d'un objet, notamment sa taille, sa légende ou sa couleur.

### Fenêtres Exécution, Variables locales et Espions

Ces fenêtres supplémentaires sont destinées au débogage de votre application. Elles ne sont disponibles que lorsque vous exécutez celle-ci dans l'environnement de développement intégré.

### Boîte à outils

Elle fournit un ensemble d'outils nécessaires au moment de la création pour disposer les contrôles sur une feuille. Outre la disposition par défaut de la boîte à outils, vous pouvez créer vos propres dispositions personnalisées en cliquant sur la commande Ajouter un onglet du menu contextuel et en ajoutant ensuite les contrôles voulus à l'onglet ainsi créé.

### L'explorateur d'objets

Un autre élément, l'explorateur d'objets, est très intéressant pour la conception d'application. On y accède par le menu « Affichage », « Explorateur d'objets » (touche F2). Il énumère les objets disponibles pour votre projet et vous permet de naviguer rapidement dans votre code. Vous pouvez recourir à l'Explorateur d'objets pour examiner les objets dans Visual Basic et d'autres applications, connaître les méthodes et propriétés disponibles pour ces objets, ainsi que pour coller des procédures de code dans votre application. Ces différents éléments seront détaillés tout au long de ce document.

## IV.4 Structure d'une application Visual Basic

Une application n'est rien d'autre qu'un ensemble d'instructions commandant à l'ordinateur d'exécuter une ou plusieurs tâches. La structure d'une application définit l'organisation de ces instructions, c'est-à-dire l'endroit où elles sont stockées et l'ordre dans lequel elles sont exécutées. Les applications simples telles que l'exemple classique « Bonjour à tous » possèdent une structure rudimentaire. Quand le code ne comprend qu'une seule ligne, l'organisation n'a guère d'importance. Toutefois, plus les applications deviennent complexes, plus elles doivent être organisées et structurées. Imaginez la confusion si le code de votre application était exécuté dans un ordre aléatoire. Outre le contrôle de l'exécution d'une application, le programmeur doit aussi se soucier de sa structure. Évaluez donc la facilité avec laquelle vous retrouvez des informations spécifiques au sein de votre application !

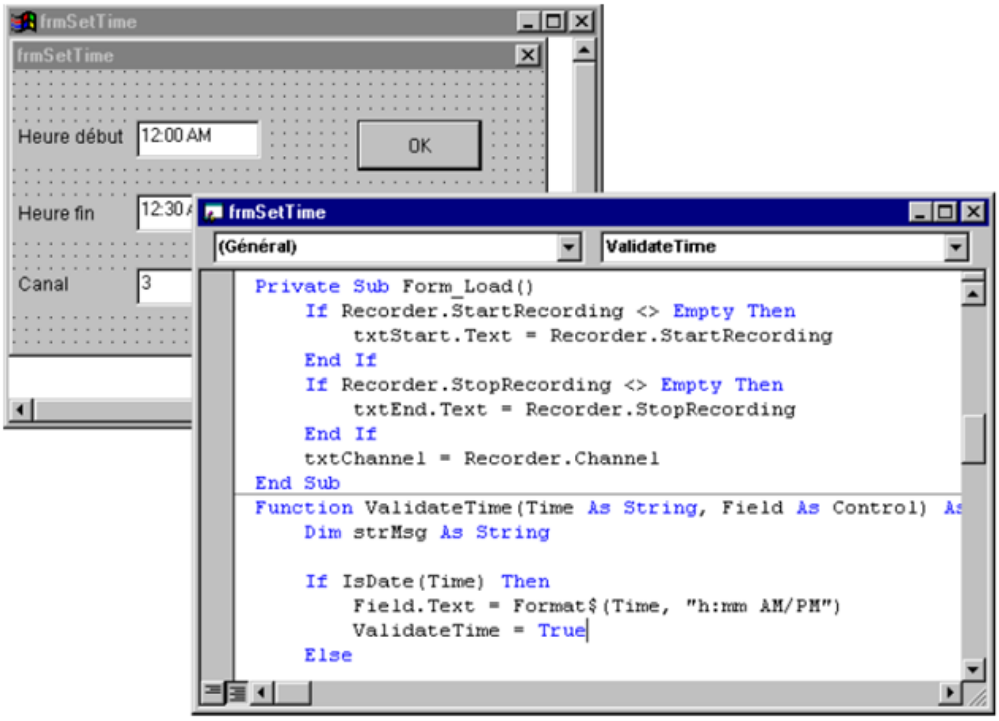



FIGURE IV.3 – Feuille et son module de feuille (code correspondant)

Chaque module de feuille contient des procédures d'événement, à savoir des sections de code dans lesquelles vous placez les instructions qui sont exécutées en réponse à des

événements spécifiques. Les feuilles peuvent aussi contenir des contrôles. Chaque contrôle d'une feuille est associé à un ensemble correspondant de procédures d'événement dans le module de feuille. Outre les procédures d'événement, les modules de feuille peuvent aussi contenir des procédures générales qui sont exécutées en réponse à l'appel d'une procédure d'événement quelconque.

Comme une application Visual Basic est basée sur des objets, la structure de son code se rapproche étroitement de sa représentation physique à l'écran. Par définition, les objets contiennent des données et du code. La feuille que vous voyez à l'écran correspond à une représentation des propriétés qui définissent son apparence et son comportement intrinsèque. À chaque feuille d'une application correspond un module de feuille connexe (dont le nom de fichier possède l'extension .FRM) contenant son code. les procédures d'événement, les modules de feuille peuvent aussi contenir des procédures générales qui sont exécutées en réponse à l'appel d'une procédure d'événement quelconque. Le code qui n'est pas associé à une feuille ou à un contrôle spécifique peut être placé dans un autre type de module, appelé module standard (.bas). Une procédure susceptible d'être exécutée en réponse à des événements dans plusieurs objets différents doit être placée dans un module standard, plutôt que de dupliquer le code dans les procédures d'événement de chaque objet (règles élémentaires de la programmation structurée). Un module de classe (.cls) permet de créer des objets susceptibles d'être appelés par des procédures internes à votre application. Alors qu'un module standard ne contient que du code, un module de classe contient à la fois du code et des données. Il ressemble plutôt à un contrôle dépourvu de représentation physique.

Le chapitre suivant explique, en traitant un exemple, comment écrire du code dans les différents éléments constitutifs d'une application. Par défaut, votre projet contient un module de feuille unique. Vous pouvez ajouter d'autres modules de feuille, de classe ou standard, si cela s'avère nécessaire.

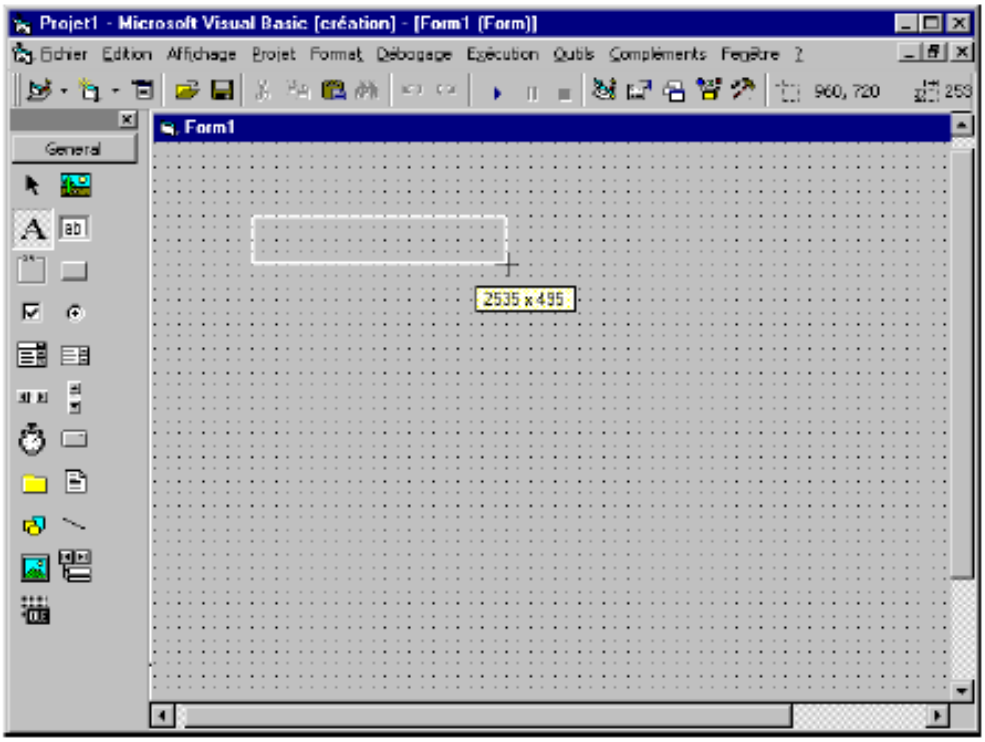



FIGURE IV.4 – Conception de froms

### IV.4.1 La création d'une application

La création d'une application dans Visual Basic implique trois étapes principales :

— Créer l'interface .

— Définir les propriétés .

— Écrire le code.

Pour mieux comprendre ces différentes étapes, les procédures suivantes vous indiquent comment créer une application simple composée d'une zone de texte et d'un bouton de commande. Lorsque vous cliquez sur celui-ci, le message « Bonjour à tous » apparaît dans la zone de texte.

### IV.4.2 Création de l'interface

Les feuilles constituent l'élément essentiel intervenant dans la création de l'interface d'une application. Elles vous permettent d'ajouter des fenêtres et des boîtes de dialogue, mais vous pouvez également les utiliser comme conteneurs d'éléments qui ne sont pas visibles dans l'interface de l'application. Par exemple, votre application pourrait contenir une feuille qui ferait office de conteneur pour des graphiques que vous envisagez d'afficher dans d'autres feuilles.

La première étape de l'élaboration d'une application Visual Basic consiste à créer les feuilles sur lesquelles sera fondée l'interface de l'application. Vous dessinez ensuite sur les feuilles les objets qui composeront l'interface. Pour cette première application, vous utiliserez deux contrôles de la Boîte à outils.

| Bouton | Contrôle |
|---|---|
| ab | Zone de texte (TextBox) |
| | Bouton de commande (CommandButton) |

FIGURE IV.5 – Controles

**Dessiner un contrôle à l'aide de la boîte à outils**

Cliquez sur l'outil correspondant au contrôle que vous souhaitez dessiner, à savoir un TextBoxet déplacer le pointeur en forme de croix jusqu'à atteindre la taille souhaitée pour le contrôle Figure ci-dessouDéfinition des propriétés L'étape suivante consiste à définir les propriétés des objets que vous avez créés. La fenêtre Propriétés (ci-dessous) permet de définir facilement les propriétés de tous les objets d'une feuille. Pour ouvrir cette fenêtre, cliquez sur la commande Fenêtre Propriétés du menu Affichage, cliquez sur le bouton Fenêtre Propriétés de la barre d'outils, ou utilisez le menu contextuel du contrôle.




FIGURE IV.6 – La fenêtre Propriétés

La fenêtre Propriétés est constituée des éléments suivants :

— Zone Objet : affiche le nom de l'objet dont vous pouvez définir les propriétés. Cliquez sur laflèche située à droite de la zone Objet de façon à afficher la liste des objets pour la feuille encours ;

— Onglets de Tri : choisissez entre une liste alphabétique des propriétés ou une vuehiérarchique divisée en catégories logiques, notamment celles relatives à l'aspect, aux policesou à la position ;

— Liste Propriétés : la colonne de gauche affiche toutes les propriétés de l'objet sélectionné.

Vous pouvez modifier et afficher les valeurs dans la colonne de droite.

**Définir les propriétés à partir de la fenêtre Propriétés**

- Dans le menu Affichage, cliquez sur Fenêtre Propriétés, ou cliquez sur le bouton Fenêtre Propriétés de la barre d'outils ;
- La fenêtre Propriétés affiche les valeurs de la feuille ou du contrôle sélectionné ;
- Dans la liste Propriétés, sélectionnez le nom d'une propriété ;
- Dans la colonne de droite, tapez ou sélectionnez la nouvelle valeur de la propriété ;
- Les propriétés énumérées possèdent une liste de valeurs prédéfinies. Vous pouvez afficher cette liste en cliquant sur la flèche vers le bas située à droite de la zone Valeurs, à moins que vous préfériez faire défiler la liste en double-cliquant sur un de éléments.

Pour l'exemple d'application « Bonjour à tous », vous devez modifier les valeurs de trois propriétés et utiliser les valeurs par défaut de toutes les autres.

| Objet | Propriété | Valeur |
|---|---|---|
| Feuille | Caption | Bonjour à tous ! |
| Zone de texte | Text | (Vide) |
| Bouton de commande | Caption | OK |

FIGURE IV.7 – Les valeurs des controles

### IV.4.3 Écriture du code

C'est dans la fenêtre Code que vous écrivez le code Visual Basic de votre application. Le code est composé d'instructions du langage, de constantes et de déclarations1. La fenêtre Code vous permet d'afficher et de modifier rapidement n'importe quelle partie du code de votre application.

**Ouvrir la fenêtre Code**

Double-cliquez sur la feuille ou le contrôle pour lequel vous souhaitez écrire du code ou dans la fenêtre Explorateur de projets, sélectionnez le nom d'une feuille ou d'un module, puis cliquez sur le bouton Code. La figure ci-dessous illustre la fenêtre Code telle qu'elle apparaît lorsque vous double-cliquez sur le contrôle Command Button, ainsi que les événements

de cette commande. Vous pouvez décider d'afficher individuellement ou simultanément les procédures dans la fenêtreCode.

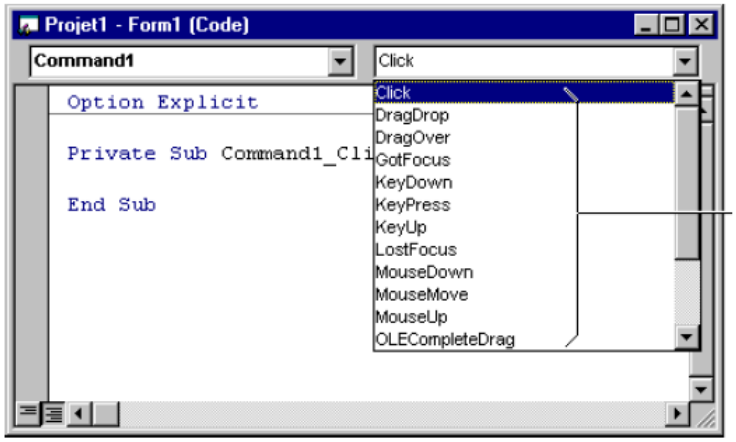



FIGURE IV.8 – Les valeurs des controles

**Afficher toutes les procédures dans la même fenêtre Code**

- Dans le menu Outils, cliquez sur Options ;
- Dans l'onglet Éditeur de la boîte de dialogue Options, activez la case à cocher située enregard de l'option « Affichage complet du module par défaut ». La case à cocher située àgauche de l'option Séparation des procédures ajoute ou supprime une ligne de séparationentre les procédures ou cliquez sur le bouton « Affichage complet du module » situé dans lecoin inférieur gauche de la fenêtre Code.

**Afficher individuellement les procédures dans la fenêtre Code**

- Dans le menu Outils, cliquez sur Options ;
- Dans l'onglet Éditeur de la boîte de dialogue Options, désactivez la case à cocher située enregard de l'option « Affichage complet du module par défaut » ou cliquez sur le bouton «Affichage de procédure » situé dans le coin inférieur gauche de la fenêtre Code.

La fenêtre Code est constituée des éléments suivants :

— Zone de liste Objet : affiche le nom de l'objet sélectionné. Cliquez sur la flèche située sur ladroite de la zone de liste pour afficher la liste de tous les objets associés à la feuille ;

— Zone de liste Procédure : énumère les procédures ou événements d'un objet. Cette zone affiche le nom de la procédure sélectionnée, en l'occurrence Click. Cliquez sur la flèche située à droite de la zone pour afficher toutes les procédures de l'objet.

**Créer une procédure d'événement**

Le code de votre application Visual Basic est divisé en blocs de petite taille appelés procédures. Les procédures d'événement, notamment celles que vous avez créées ici, contiennent du code qui est exécuté au moment où survient un événement (notamment lorsqu'un utilisateur clique sur un bouton). Une procédure d'événement d'un contrôle combine le nom effectif du contrôle (spécifié dans la propriété Name), un trait de soulignement (_) et le nom de l'événement. Par exemple, si vous souhaitez qu'une procédure d'événement soit invoquée lorsque l'utilisateur clique sur un bouton de commande nommé Command1, utilisez la procédure Command1_Click.

— Dans la zone de liste Objet, sélectionnez le nom d'un objet dans la feuille active. (La feuilleactive n'est autre que la feuille en cours). Pour cet exemple, cliquez sur le bouton decommande Command1 ;

— Dans la zone de liste Procédure, sélectionnez le nom d'un événement de l'objet sélectionné.Dans notre exemple, la procédure Click est déjà sélectionnée, car il s'agit de la procédure par défaut pour un bouton de commande. Notez qu'un modèle de la procédure d'événement est à présent affiché dans la fenêtre Code ;

— Tapez le code suivant entre les instructions Sub et End Sub :

```
Text1.Text = "Bonjour à tous !"
La procédure d'événement devrait se présenter comme suit :
Private Sub Command1_Click ()
   Text1.Text = "Bonjour à tous !"
End Sub
```

FIGURE IV.9 – Code en VB 6.0

Vous remarquerez ici que le code modifie simplement la propriété Text du contrôle nommé Text1 en « Bonjour à tous ». La syntaxe de cet exemple prend la forme object. property dans laquelle Text1 est l'objet et Textla propriété. Cette syntaxe vous permet de modifier les valeurs des propriétés de n'importe quelle feuille ou contrôle en fonction des événements qui surviennent pendant l'exécution de votre application.

### IV.4.4 Exécution de l'application

Pour exécuter l'application, cliquez sur Exécuter dans le menu Exécution, ou cliquez sur le bouton Exécuter de la barre d'outils, ou encore appuyez sur F5. Cliquez sur le bouton de commande que vous avez créé sur la feuille et vous verrez apparaître « Bonjour à tous » dans la zone de texte.

# Chapitre V

# Conception de S.U.A.S v1.0

Présentation générale : le programme SUAS v 1 .O(soil under axisymetrique solicitation) et un petit programme de calcul des contraint dans le sol cas des problèmes axisymétriques) Basée sur la méthode des éléments finis comme le montre la figure se de sous :

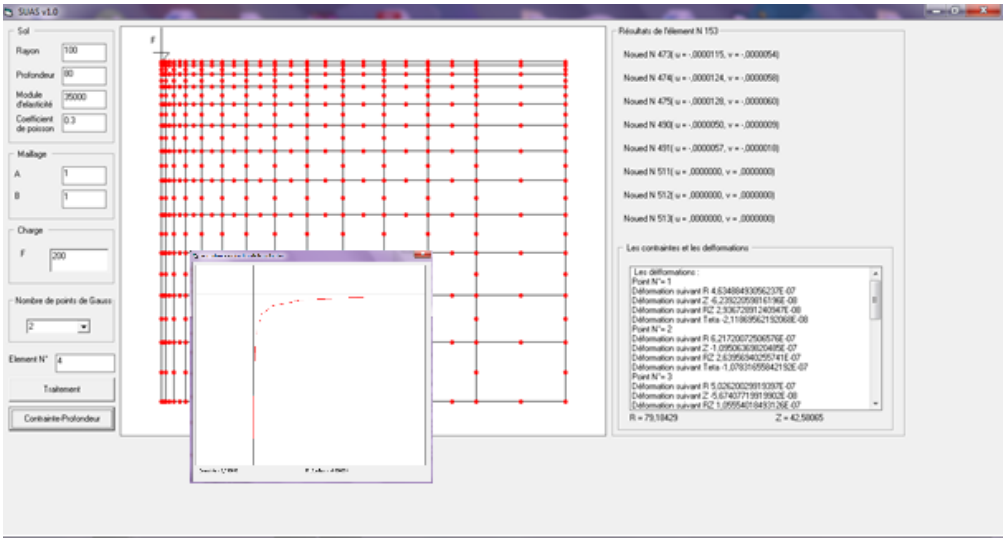



FIGURE V.1 – interface de programme SUAS v 1.0

ce programme et développée avec le fameux logicielle de programmation visuel basic 6 de l'entreprise américain (Microsoft)

# V.1 les entrés

Comme toute programme de da calcul le programme SUAS v 1 .O a des champs d'entrée les donnée qui s'appelle l'input dans (profondeur de sol .rayon des cercle de sol .module de Young. Coefficients de poisson.nombre de point de gauss).

## V.1.1 Propriété géométrique

La profondeur et le rayon des sols chois Sant entre avec les champs sur l'interface des programme commenous montreles figure si de sous.

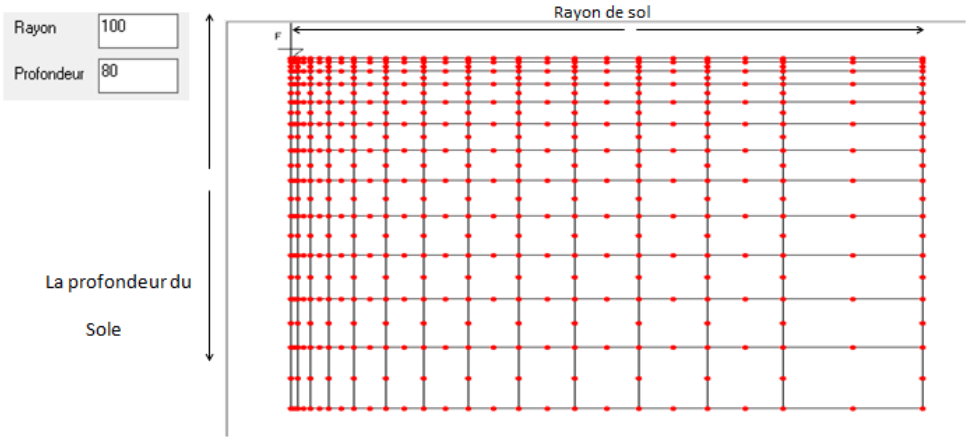



FIGURE V.2 – 18manuel utilisation SUAS v 1.0

## V.1.2 Propriété des matériaux

Les propriétés des matériaux au de sol sont défini entre dans le programme par le champ comme ne montre le figure si de sous .

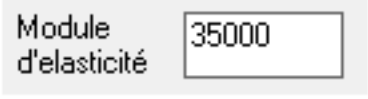



FIGURE V.3 – Module d'élasticité

ce champ et utiliser pour entre le module d'élasticité (module de Young de chaque sol)




FIGURE V.4 – Coefficient de poisson

ce champ et utiliser pour le coefficient de poisson

### V.1.3 Propriété des maillages

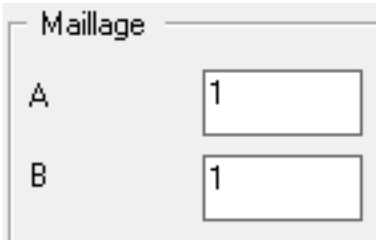



FIGURE V.5 – Paramètres des maillages

Les deux coefficients A, B sont les paramètres des maillages au :

— A : est la longueur de premier élément quadrangulaire

— B : est la largeur de premier élément quadrangulaire

### V.1.4 La charge

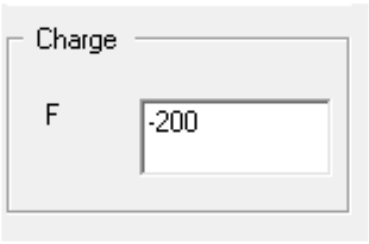



FIGURE V.6 – La charge concentrée

La charge concentrée applique sur le sol

### V.1.5 Nombre de point de gauss

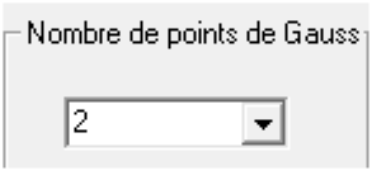



FIGURE V.7 – Nombre de point de gauss

Nombre de point de gauss utiliser.

## V.2 les sortie

### V.2.1 Déplacement

Elle donne pour chaque élément quadrangulaire les deux déplacement vertical et horizontale $(u, v)$ pour chaque nœuds.




FIGURE V.8 – Déplacements vertical et horizontale

### V.2.2 Déformation

Elle donne les quatre déformations $\varepsilon_r \varepsilon_z \varepsilon_\theta \varepsilon_{rz}$

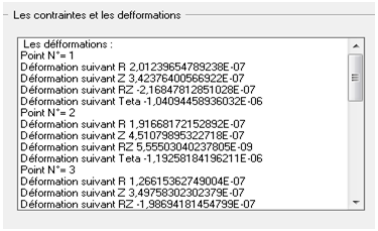



FIGURE V.9 – les quatre déformations

### V.2.3 Les contraintes

Les quatre contraintes $(\sigma_r, \sigma_z, \sigma_\theta, \sigma_{rz})$ Applique au point de gausse Les sorti des programme sont les déplacements les déformations et les contrainte




FIGURE V.10 – Les quatre contraintes

## V.3 La programmation des éléments finis

### V.3.1 L'algorithme

#### Numerotation local et globale des nœuds

dans la méthode des éléments finis la numérotation du nœud a une grande importance car cette numérotation et choisi selon la discrétisation (élément triangulaire au quadrangulaire) selon le nombre des nœuds dans l'élément de discrétisation donc chaque nœud a deux numérotation un numérotation par rapport à l'élément qui il contient (numérisation local) et une numérotation par rapport au Élément real (numérotation globale) le choix de la méthode de numérotation globale défini la forme de matrice de rigidité globale, donc un bonne choix de numérotation déminé le nombre des opérations et le temps pour résoudre système l'équation $\{F\} = [K]\{U\}$

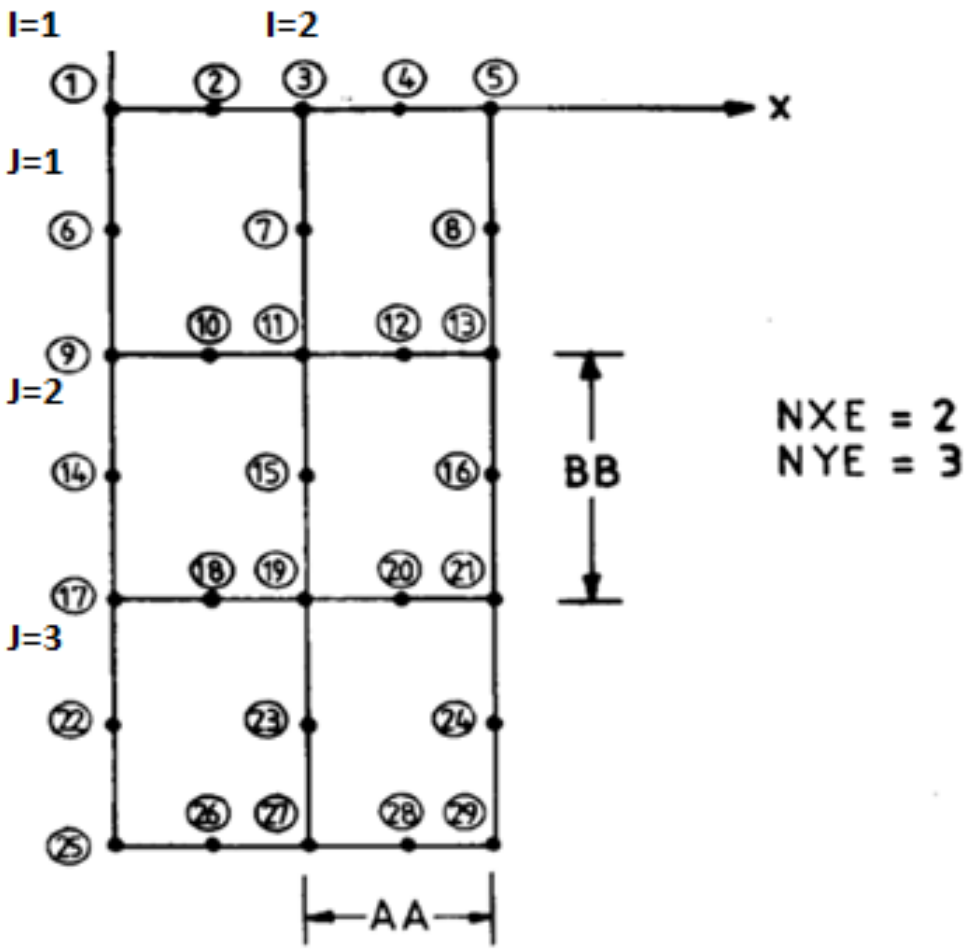



FIGURE V.11 – numérotations locales et globales

Don notre exemple au on a choisi un élément quadrangulaire a huit nœud au la numérotation et dans le sens des X cette numérotation et forme avec les boucle suivant :

```
NE = 0
For i = 1 To nex
For j = 1 To ney
NE = NE + 1
ELE(NE, 1) = (i - 1) * (3 * ney + 2) + 2 * j + 1 ' 3
ELE(NE, 2) = ELE(NE, 1) - 1
```

ELE(NE, 3) = ELE(NE, 2) - 1

ELE(NE, 4) = (i - 1) * (3 * ney + 2) + 2 * ney + j + 1

ELE(NE, 5) = (i) * (3 * ney + 2) + 2 * j - 1

ELE(NE, 6) = ELE(NE, 5) + 1

ELE(NE, 7) = ELE(NE, 6) + 1

ELE(NE, 8) = ELE(NE, 4) + 1

Next

Next

Stocker dans la matrices ELE(NE, k)

- $NE$ :le numéro de l'Elément (écrit Dans le figure en chiffre romain,I, II,III,IV ...)
- $K = 1, 2 \ldots 8$ le numérotation local de nœud que vous calcul sont numérotation globale
- ($NEX = 3$) nombre des éléments dans le sens des X Et ($NEY = 3$) nombre des éléments le sens Y

**matrice coordonne des nœuds (coord)**

La matrice coordonnée est la matrice qui stock les coordonne des nœuds selon la discrétisation choisi dans notre cas élément quadrangulaire au la discrétisation est très série au endroit les grandes variations des contrainte (point d'application des charge) et large de plus en s'éloigne des zone d'application des charge La matrice coordonnes et de démontions NN X2 Coord(ele,2) AU le premier indice représente le nombre de nœud (numérotations globale) Le deuxième indice représente le numéro de démontions (1 pour x ,2 pour Y cas d'état plan)

**Vecteur fonction de forme FUN**

$$FUN = \begin{bmatrix} \frac{-(1-\varepsilon)(1-\eta)(1+\varepsilon+\eta)}{4} \\ \frac{(1-\varepsilon^2)(1-\eta)}{2} \\ \frac{-(1-\varepsilon)(1-\eta)(1-\varepsilon+\eta)}{4} \\ \frac{(1-\eta^2)(1+\varepsilon)}{2} \\ \frac{-(1+\varepsilon)(1-\eta)(1-\varepsilon-\eta)}{4} \\ \frac{(1-\varepsilon^2)(1+\eta)}{2} \\ \frac{-(1-\varepsilon)(1+\eta)(1+\varepsilon-\eta)}{4} \\ \frac{(1-\eta^2)(1-\varepsilon)}{2} \end{bmatrix}$$

**Matrice dérivée des fonctions de forme DER dans le repère locale ($\varepsilon_\eta$)**

$$DER = \begin{bmatrix} \frac{\partial FUN^T}{\partial \varepsilon} \\ \frac{\partial FUN^T}{\partial \eta} \end{bmatrix}$$

**Matrice jacobinne jac**

$$x = x + DER(ii, f) \times COORD(ELE(el, f), ji)$$

**Matrice inverse jacobinne $jac^{-1}$**

**a.1 Matrice dérivée des fonctions de forme DERIV dans le repère globale(x y)**

$$DERIV(ii, ji) = jac1(ii, f) * DER(f, ji)$$

# Chapitre VI

# Comparaison entre la méthode de boussinesq et SUAS v1.0

## VI.1 Les exemples de calcules

**Exemple N01** En prend l'argile qui a les caractéristiques suivant E module de Young =1000

Coeffiient de poisson =0.35

La contrainte verticale due à une charge concentre sur un massif de sol d'un point de cordonne (R, Z) est donne par la loi de Boussinesq .

$$\sigma_z = \frac{3pz^3}{2\pi R^5}$$

- $\sigma_z$ : la contrainte vertical.
- $Z$ : profondeur de point donne.
- $R$ : distance horizontale séparant le point que on veut lui calculer sont contraint et le point d'application des force.
- $P$ :la charge concentre applique sur le massif de sol On prend une charge et égale à 80 N Pour un point qui a les coordonne suivantes (78.09 ;21.24)

D'après les calculs on obtient $\sigma_z = 0.00047N/m^2$. Maintenant en calcul avec le programme S U A S on obtient les résultats suivant

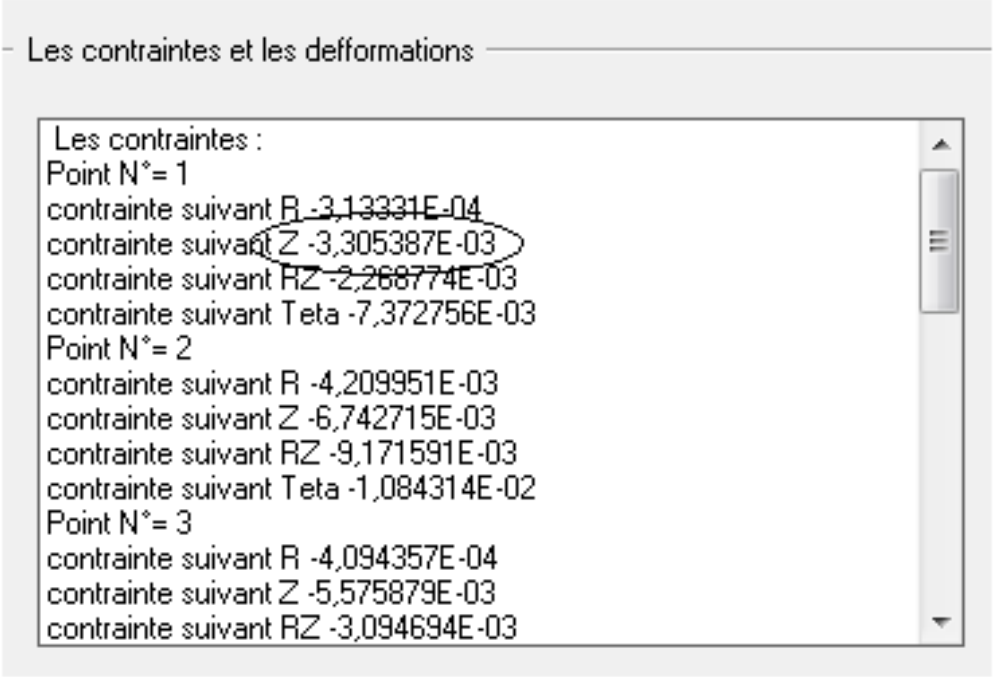



FIGURE VI.1 – Les resultats du S.U.A.S

La contrainte verticale $\sigma_z = 0.0033N/m2$ Les contrainte en fonction de la profondeur

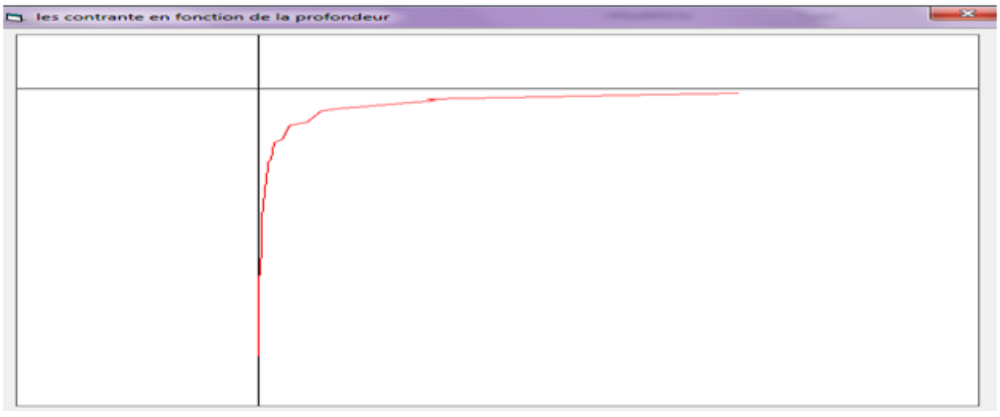



FIGURE VI.2 – Le graphe de contrainte en fonction de la profondeur

**Exemple N02** En prend un sol sableux qui a les caractéristique suivant :

E : module de Young =40000

v ; Coefficient de poisson =0.35

Pour un point qui a les cordonne suivante (28.24 ;15.74) D'après le calcul avec la méthode de Boussinesq pour une charge de 80 kp On obtient le résultat suivant $\sigma_z = 0.031N/m2$

Le calcule avec le logiciel S U A S donne les résultats suivant

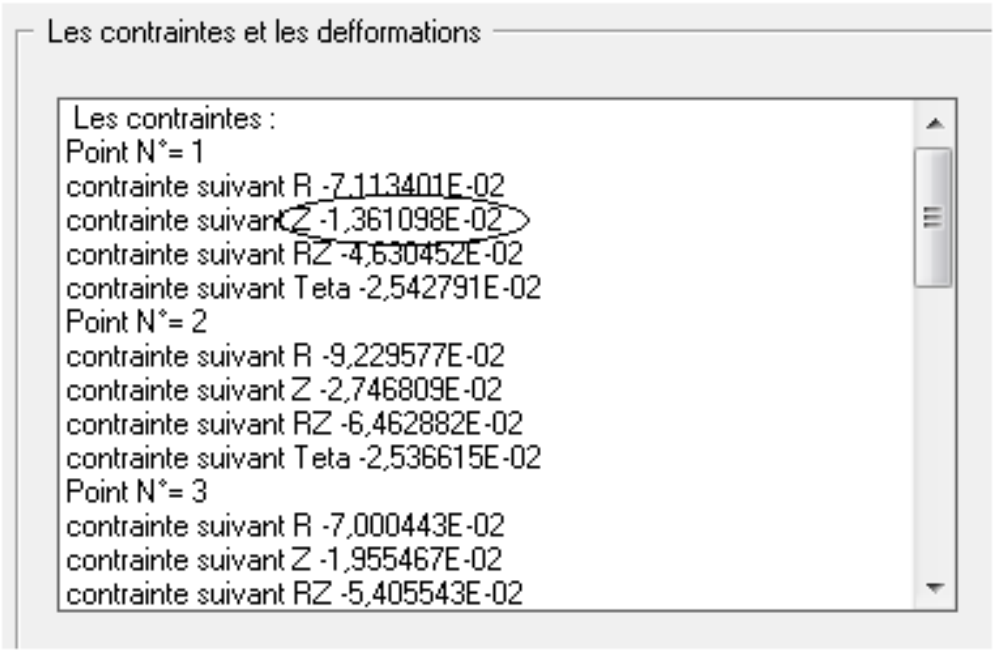



FIGURE VI.3 – Les resultats du S.U.A.S

La contrainte verticale $\sigma_z = 0.013N/m^2$

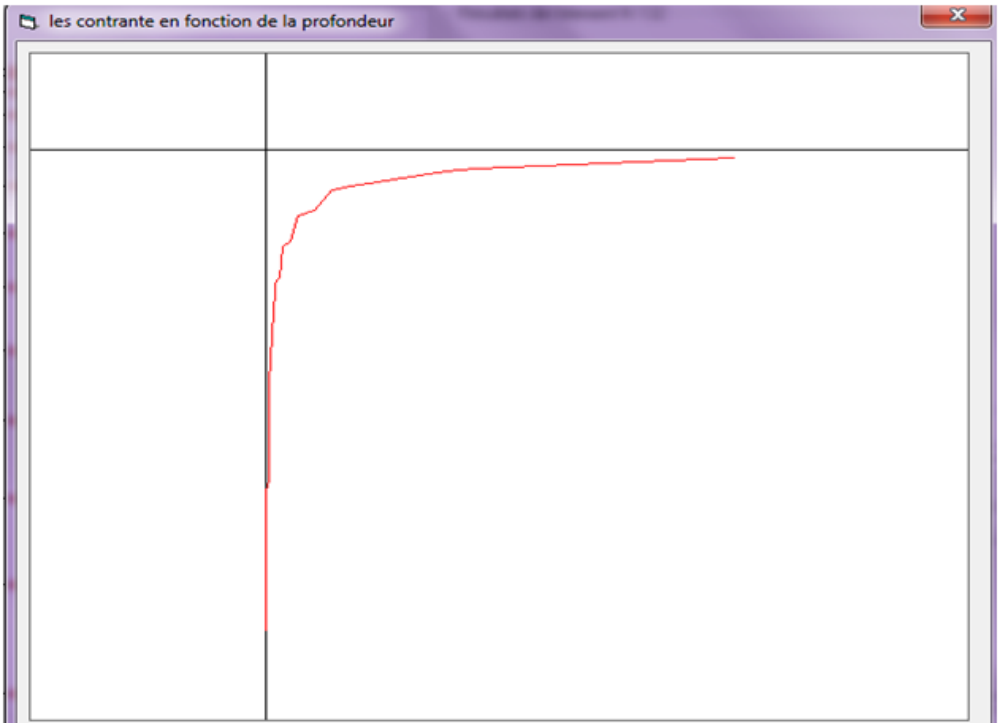



FIGURE VI.4 – Le graphe de contrainte en fonction de la profondeur

**Exemple N03** En prend un sol de limon qui a les caractéristique suivant :

E : module de Young =20000

v : Confisions de poisson =0.33

Pour un point qui a les cordonne suivante (36.22 ;10.40)

D'après le calcule avec la méthode de Boussinesq pour une charge de 80 kp

On obtient le résultat suivant $\sigma_z = 0.0025 N/m^2$

Le calcule avec le logicielle donne les résultats suivant

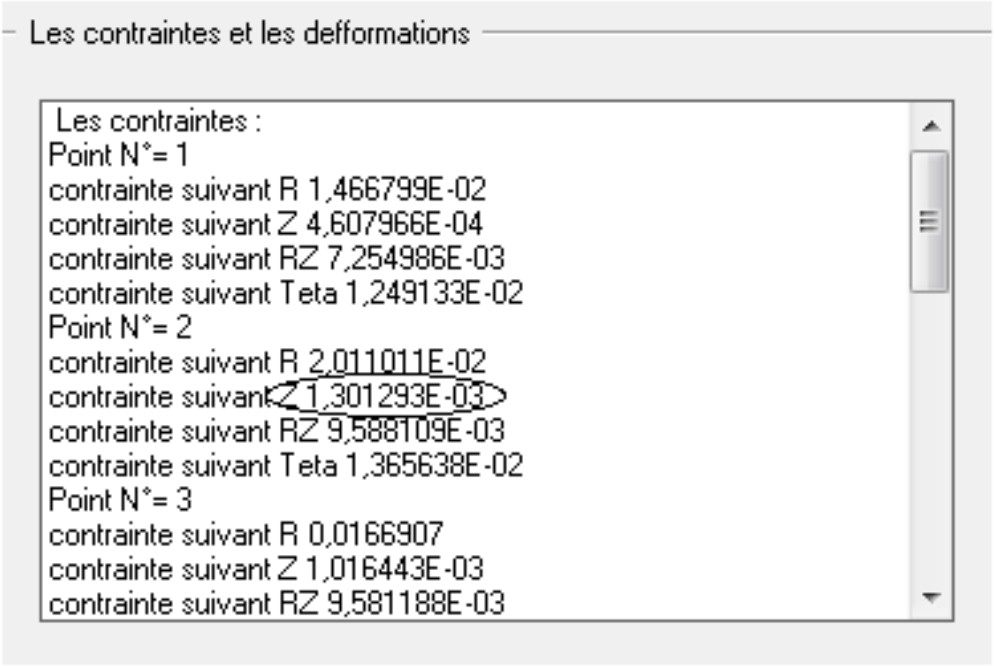



FIGURE VI.5 – Les resultats du S.U.A.S

La contrainte verticale $\sigma_z = 0.013N/m^2$

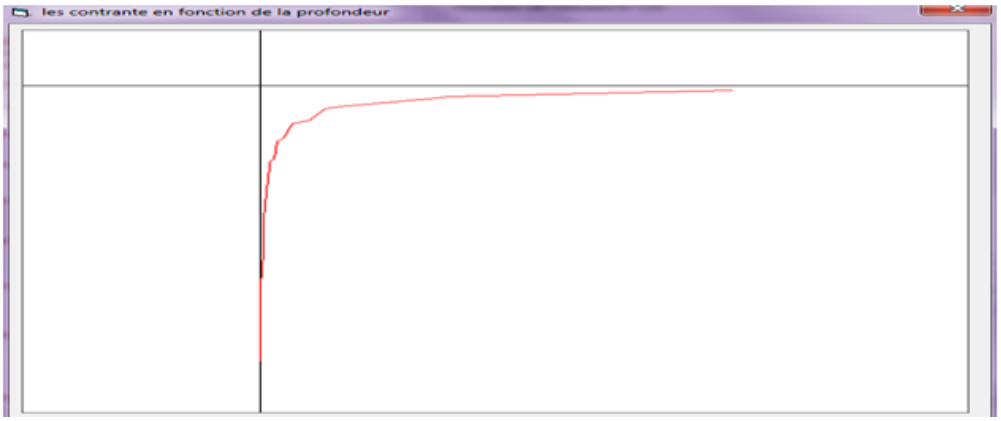



FIGURE VI.6 – Le graphe de contrainte en fonction de la profondeur

**Exemple N04** On change la valeur de la charge et en reparte le calcule pour tous les type de sol précédent Pour le sol argileux les résultats obtenus avec une charge F= 200 N

ou point des cordonne (78.27 ;21.24)

On utilisant la méthode de Boussinesq en va trouver $\sigma = 0.00012N/m^2$ Le calcule avec le logicielle SUAS donne les résultats suivant




FIGURE VI.7 – Les resultats du S.U.A.S

La contrainte verticale $\sigma_z = 0.00305N/m^2$

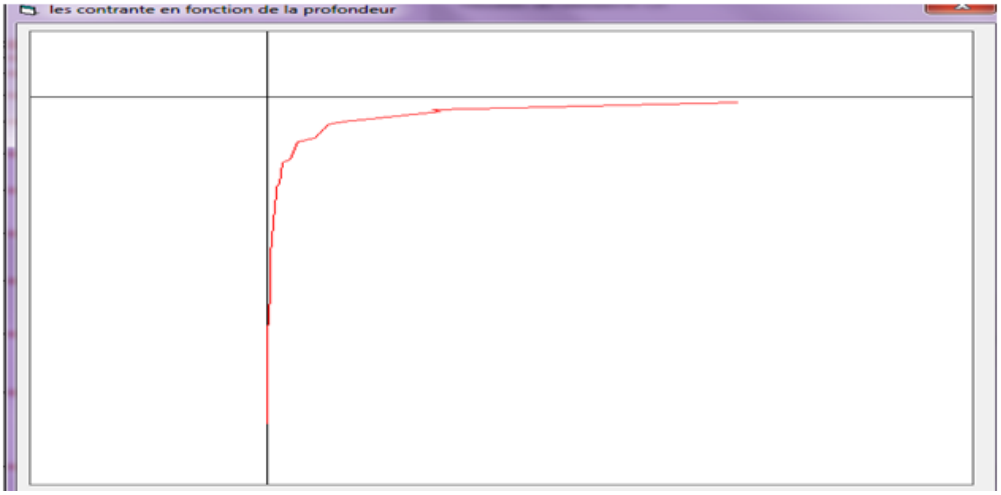



FIGURE VI.8 – Le graphe de contrainte en fonction de la profondeur

**Exemple N05** Pour le sol sableux avec les mêmes conditions pour un point (78.27 ; 28.12) Dans la méthode de Boussinesq on obtient le résultat suivant $\sigma_z = 0.000723N/m^2$ Le calcule avec le logicielle SUAS donne les résultats suivant

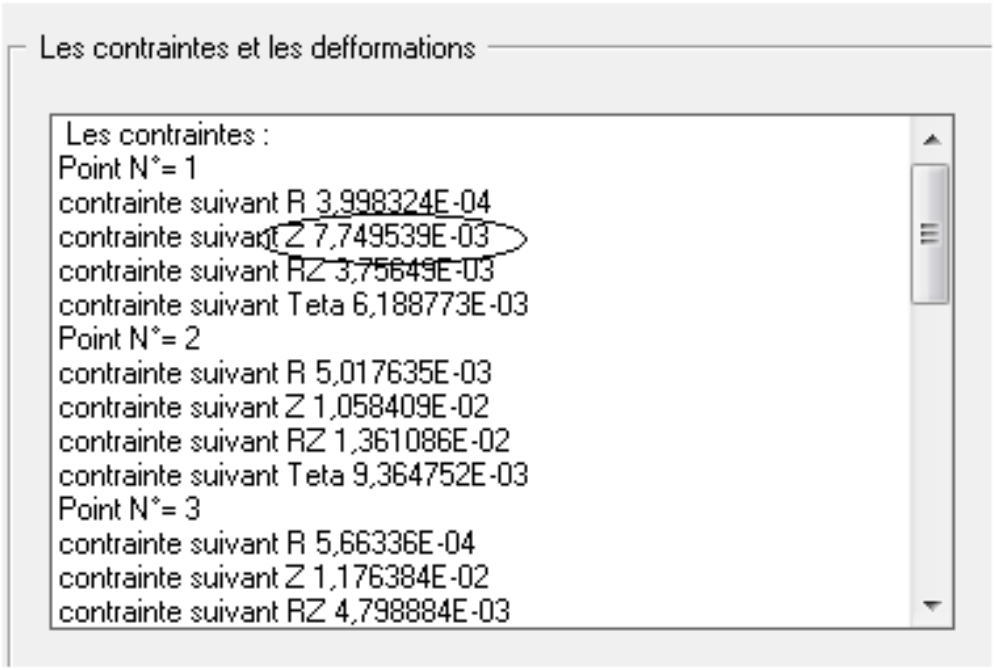



FIGURE VI.9 – Les resultats du S.U.A.S

La contrainte verticale $\sigma_z = 0.00740N/m^2$

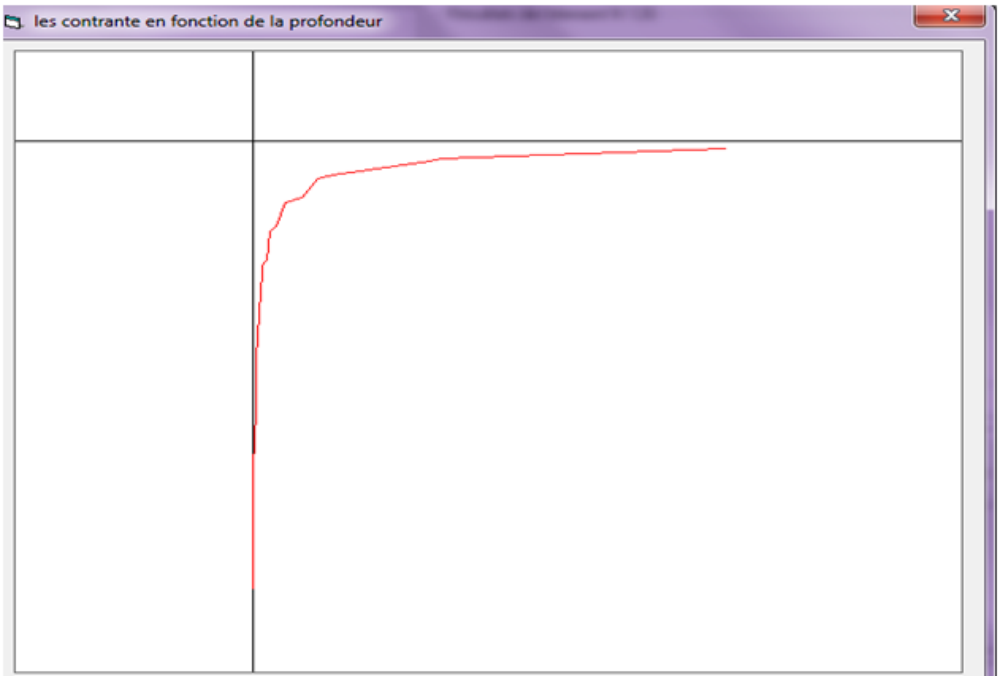



FIGURE VI.10 – Le graphe de contrainte en fonction de la profondeur

**Exemple N06** Pour le sol limon avec les mêmes conditions pour un point (77.73 ; 24.86) Dans la méthode de Boussinesq on obtient le résultat suivant :

$\sigma_z = 0.000517 N/m^2$ Le calcule avec le logicielle SUAS donne LES résultats suivant

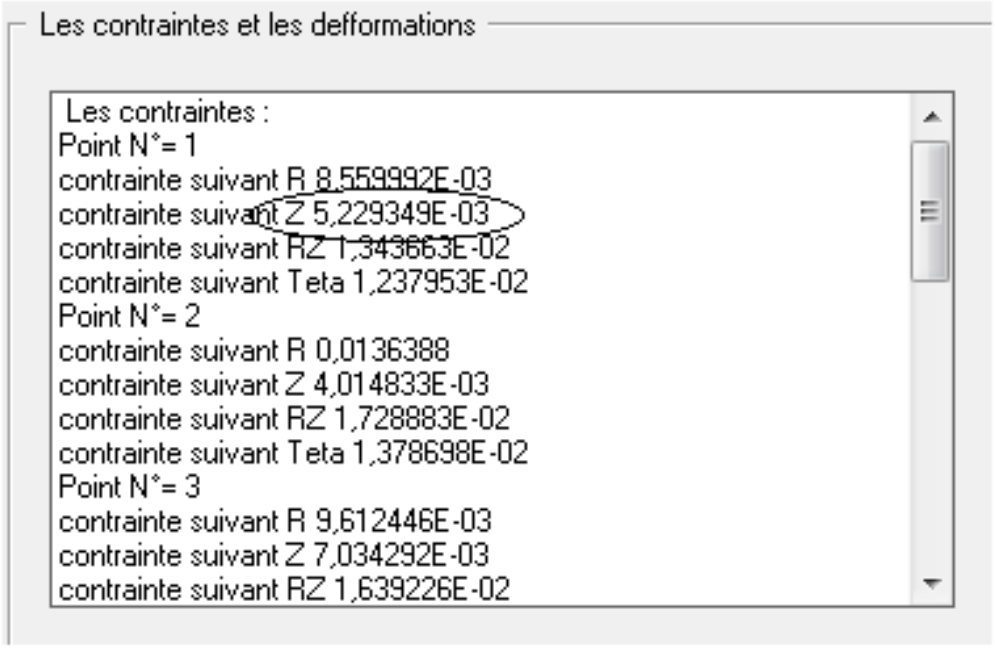



FIGURE VI.11 – Les resultats du S.U.A.S

La contrainte verticale $\sigma_z = 0.00522 N/m^2$

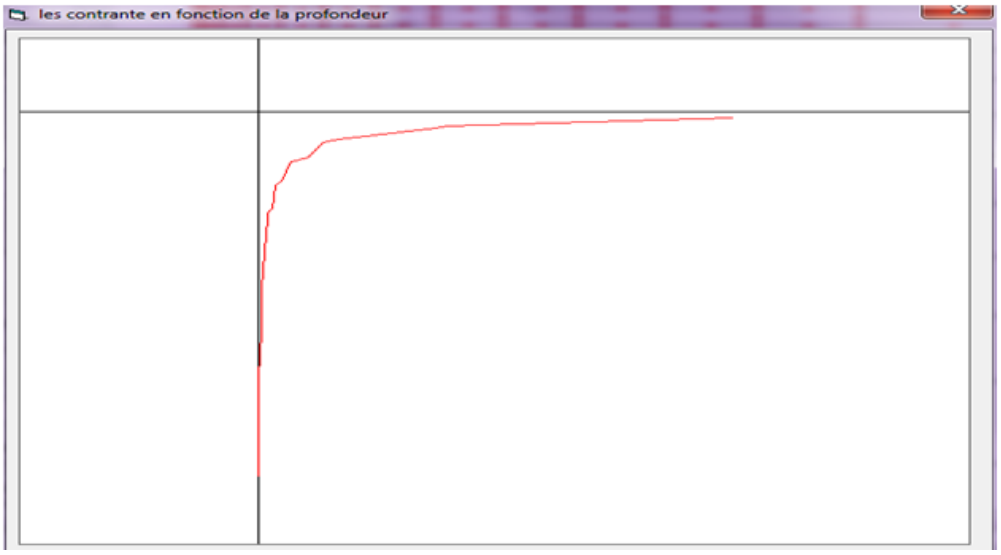



FIGURE VI.12 – Le graphe de contrainte en fonction de la profondeur

En résume les résultats calculés précédemment dans le tableaux suivant pour la charge de $200N$

| Type de sol | Module de Young | Coefficient de poisson | Profondeur z | Rayon R | Contraint vertical calculer avec méthode de Boussinesq | Contraint vertical calculer avec logicielle suas v1.o |
|---|---|---|---|---|---|---|
| Argile | 10000 | 0 .35 | 21.24 | 78.27 | 0.00012 | 0.00305 |
| Sable | 40000 | 0.3 | 28.12 | 78.27 | 0.000723 | 0.00740 |
| Limon | 20000 | 0.33 | 24.86 | 77.73 | 0.000517 | 0.00522 |

TABLE VI.1: Résume des résultats calculés

Pour la charge de 80 $N$.

| Type de sol | Module de Young | Coefficient de poisson | Profondeur z | Rayon R | Contraint vertical calculer avec méthode de Boussinesq | Contraint vertical calculer avec logicielle suas v1.o |
|---|---|---|---|---|---|---|
| Argile | 10000 | 0 .35 | 21.24 | 78.09 | 0.00047 | 0.0033 |
| Sable | 40000 | 0.3 | 15.74 | 28.24 | 0.031 | 0.013 |
| Limon | 20000 | 0.33 | 10.40 | 36.22 | 0.0025 | 0.0049 |

TABLE VI.2: Résume des résultats calculés

## VI.2 Interprétationdes résultats

D'aprèsles résultats, on peut dire que pour la charge 80 les résultats de SUAS V1 .0etles résultats de la méthode de Boussinesq sont comparable par contre pour la charge de 200 KN les résultats sont incomparable ; cette différence est liée au comportement du sol appliqué dans notre programme (loi de Hooke) par contre la méthode de Boussinesq utilise les parametres geometrique R et Z et ne prend pas les comportements du sol en consideration.

## VI.3 Conclusion générale

L'occasion de la préparation du sujet de master nous permettre de toucher la magie de la méthode des éléments finis, en effet, l'objectif de ce travail est de comprendre l'application et la programmation de la méthode des éléments finis dans le domaine géotechnique, et on peut dire que on a atteindre cet objet à travers de SUAS v1.0 , ce programme ne peut en aucune cas remplacé ou bien conque les programmes commercialisés dans le marché tel que PLAXIS, ABACUS . . .mais il est considéré comme un programme pédagogique qui nous permettre de maitriser la méthode des éléments finis La comparaison entre les méthodes analytique et les méthodes numériques est très difficiles pour plusieurs raisons . Pour notre cas la comparaison des résultats est faisable si les charges a une petite valeure (puisque le comportement du sol est linéaire) par contre pour les charges a une grande valeurs les résultats sont incomparable ; cette différence est liée au comportement du sol appliqué dans notre programme (loi de Hooke) par contre la méthode de Boussinesq n'utilise que les paramétres géométriques du sol .

Avec le même code et avec des petits changements on peut modéliser des autres problèmes tels que les pieux chargé axialement. Ces changements sont liés au maillage et l'affectation des propriétés physiques de chaque élément (module d'élasticité et le coefficient de poisson)

# Appendices

# Annexe A

# Subroutune numérotation ELE et coordonnée coord

```
For i = 1 To NEX
FOR J= 1 To NEY
ELE(NE, 1) = (i - 1) * (3 * NEY + 2) + 2 * j + 1 ' 3
ELE(NE, 2) = ELE(NE, 1) - 1
ELE(NE, 3) = ELE(NE, 2) - 1
ELE(NE, 4) = (i - 1) * (3 * NEY + 2) + 2 * NEY + j + 1
ELE(NE, 5) = (i) * (3 * NEY + 2) + 2 * j - 1
ELE(NE, 6) = ELE(NE, 5) + 1
ELE(NE, 7) = ELE(NE, 6) + 1
ELE(NE, 8) = ELE(NE, 4) + 1
COORD (ELE(NN, 1) ,1) =x_step *(J- 1)
COORD (ELE (NN, 2 ) ,1) =x_step *(J- 1)
COORD (ELE (NN, 3) ,1) = x_step *(J- 1)
COORD (ELE (NN, 5 ) ,1) =x_step *J
COORD (ELE (NN, 6 ) ,1) = x_step *J
COORD (ELE (NN, 7) ,1) = x_step *J
```

```
COORD (ELE (NN, 4) ,1) = 0 ,5(coord(ELE(NN, 3) ,1) + coord(ELE(NN, 5) ,1) )
COORD ( (ELE (NN, 8),1) = 0 ,5(coord(ELE(NN, 7) ,1) + coord(ELE(NN, 1) ,1) )
COORD (ELE (NN, 1) ,2) =Y_step *(I )
COORD (ELE (NN, 8) ,2) =Y_step *(I )
COORD (ELE (NN, 7) ,2) = bb*(I )
COORD (ELE (NN,3) ,2) = bb*(I- 1 )
COORD (ELE (NN, 4 ) ,2) = bb*(I- 1 )
COORD (ELE (NN, 5) ,2) = bb*(I- 1 )
COORD (ELE (NN, 2) ,2) = 0 ,5 (coord(ELE (NN, 1) ,2 ) + coord(ELE (NN, 5) ,1) )
COORD (ELE (NN, 6) ,2) = 0 ,5 (coord(ELE (NN, 5) ,2 ) + coord(ELE (NN, 7) ,2) )
NEXT
```

# Annexe B

# Subroutine matrice D

```
Private Sub fmdrad()
Dim v1 As Double
Dim c As Double
v1 = 1 - v
c = e / ((1 + v) * (1 - 2 * v))
DEE(1, 1) = v1 * c
DEE(2, 2) = v1 * c
DEE(3, 3) = 0.5 * c * (1 - 2 * v)
DEE(4, 4) = v1 * c
DEE(1, 2) = v * c
DEE(2, 1) = v * c
DEE(1, 3) = 0
DEE(3, 1) = 0
DEE(1, 4) = v * c
DEE(4, 1) = v * c
DEE(2, 3) = 0
DEE(3, 2) = 0
DEE(2, 4) = v * c
```

```
DEE(4, 2) = v * c
DEE(4, 3) = 0
DEE(3, 4) = 0
End Sub
```

# Annexe C

# Subroutine de vecteur function de form FUN et matrice derive DER

```
Private Sub formln(i As Integer, j As Integer)
Dim eta, xi, etam, etap, xim, xip As Double
eta = SAMP(i, 1)
xi = SAMP(j, 1)
etam = 0.25 * (1 - eta)
etap = 0.25 * (1 + eta)
xim = 0.25 * (1 - xi)
xip = 0.25 * (1 + xi)
FUN(1) = 4 * xim * etam*(- xi- eta -1 )
FUN(2) = 32 * etam* xim * etap
FUN(3) =4 * etap * xim *(- xi+ eta -1 )
FUN(4) = 32* xim * xip * etap
FUN(5) =4* etap* xip *(xi+ eta -1 )
FUN(6) = 32 * etap* xip *etam
FUN(7) =4* xip*etam *(xi- eta -1 )
FUN(8) =32*xim* xip* etam
```

```
DER(1, 1) = etam* (2* xi+ eta )
DER(1, 2) = -8* etam* etap
DER(1, 3) = etap* (2* xi- eta )
DER(1, 4) =-4*etap* xi
DER(1, 5) =etap* (2* xi+ eta )
DER(1, 6) =8* etap *etam
DER(1, 7) =etam* (2* xi- eta )
DER(1,8) =-4* etam * xi
DER(2, 1) =xim* (xi+2* eta )
DER(2, 2) =-4* xim* eta
DER(2, 3) =xim* (2* eta-xi )
DER(2, 4) =8* xim * xip
DER(2, 5) =xip*(xi+2* eta )
DER(2, 6) =-4*xip * eta
DER(2, 7) =xip*(2* eta-xi )
DER(2, 8) = -8* xim * xip
End Sub
```

# Annexe D

# Subroutine l'élimination de gausse

```
Private Sub triangularisation()
Dim Triangular_A() As Double
Dim temporary_1 As Double
Dim multiplier_1 As Double
Dim line_1 As Long
ReDim Triangular_A(Nq, Nq + 1)
For i = 1 To Nq
For j = 1 To Nq
Triangular_A(i, j) = K(i, j)
Next
Next
'Dim temporary_1 As Double
For N = 1 To Nq
Triangular_A(N, Nq + 1) = Force(N)
Next N
'Triangularize the matrix
For Kg = 1 To Nq - 1
'Bring a non-zero element first by changes lines if necessary
```

```
If Triangular_A(Kg, Kg) = 0 Then
For N = Kg To Nq
If Triangular_A(N, Kg) ¡¿ 0 Then line_1 = N : Exit For 'Finds line_1 with non-zero element
Next N
'Change line k with line_1
For m = Kg To Nq + 1
temporary_1 = Triangular_A(Kg, m)
Triangular_A(Kg, m) = Triangular_A(line_1, m)
Triangular_A(line_1, m) = temporary_1
Next m
End If
'For other lines, make a zero element by using :
'Ai1=Aij-A11*(Aij/A11)
'and change all the line using the same formula for other elements
For N = Kg + 1 To Nq
If Triangular_A(N, Kg) ¡¿ 0 Then 'if it is zero, stays as it is
multiplier_1 = Triangular_A(N, Kg) / Triangular_A(Kg, Kg)
For m = Kg To Nq + 1
Triangular_A(N, m) = Triangular_A(N, m) - Triangular_A(Kg, m) * multiplier_1
Next m
End If
Next N
Next Kg
For N = 1 To Nq
For m = 1 To Nq
K(m, N) = Triangular_A(m, N)
Next
Next
```

```
'Assign values from array b
For N = 1 To Nq
Force(N) = Triangular_A(N, Nq + 1)
Next N
End Sub
```

# Bibliographie


[1] *I. M .Smith and D.V.Griffith univercity of manchesterprogramming the finite element*

[2] *GÉOTECHNIQUE 1 Cours ¬ Chapitres 3 institut national des sciences applique de Toulouse*

[3] *Robert D.CookDavid S. Markus Micheal E .PLeshaRobert j .WittConcept and applications of finite element analyse*

[4] *Michel CazenaveMéthode des éléments finisApproche pratiqueen mécanique des structures*

[5] *Bakhti Rachid(2014) mimoiremagister analyse par elementsfinisdespieux isoles sous sollicitations verticales et horizontales prise en compte de l'interface sol /pieu*

[6] *Berga Abdelmadjid (202004-03) élément de Mécanique des sols centre universitaire de bechar*

[7] *Herve Oudin méthode des élément finie*

[8] *Méthode des éléments finisen élasticité linéaire*

[9] *A.KHANENE :Méthode des élément finis ,Enonce de base .office des publication Universitaire1997*

[10] *M K BERRAH, COUR DE MEF 1 ERE anne Post graduation ENTP 1994 1995*

[11] *G WARZEE :méthode des élément finie Presse univercitaire de bruxelles 3 eme edition 1989 -1990*