

وزارة التربية الوطنية
MINISTERE DE L'EDUCATION NATIONALE

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT GENIE MECANIQUE

المدرسة الوطنية المتعددة التخصصات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDES

SUJET

RESOLUTION DES PROBLEMES
D'ELASTOSTATIQUE TRIDIMENSIONNELS PAR
LA METHODE DES EQUATIONS INTEGRALES
DIRECTE

Proposé par :

Y . BELKACEMI

Etudié par :

KEBIR HOCINE

Dirigé par

Y . BELKACEMI

PROMOTION

JUILLET 1993

DEDICACES

A mes très chers parents

A mon très cher frère AHMED

A toute ma famille

Et à tous ceux qui sont chers à mon coeur

Je dédie ce modeste travail

KEBIR HOCINE



REMERCIEMENTS

Je tiens à adresser ma profonde reconnaissance à mon promoteur Y. BELKACEMI pour ses conseils et son suivi durant toute la période de préparation de mon projet de fin d'étude .

Je remercie également F.BELBLIDIA dont l'aide m'a été précieuse pour la réalisation de ce mémoire .

Je tiens en fin à exprimer ma profonde gratitude à tous ceux qui ont contribué à ma formation, notamment :

M . GAHMOUSSE
M . RECHAK
M . DJELLOULI

H. KEBIR

ملخص :

الهدف من هذه الدراسة هو ايجاد حقل الانتقالات وحقل الاجهادات داخل جسم متجانس ، ثلاثي الابعاد ، ذو تصرف خطي مرن و ذلك باستعمال طريقة المعادلات التكاملية المباشرة

RESUME :

Le but de cette étude est de déterminer le champ des déplacements et des contraintes dans un corps homogène , élastique et linéaire , en trois dimensions , par la méthode des équations intégrales directe.

ABSTRACT :

The object of this work is to found the field of displacements and the field of strees in an homogenous elastic and linear body , in three dimensions , using the direct boundary method .

SOMMAIRE

CHAPITRE 1 :

GENERALITES SUR LES METHODES DES EQUATIONS INTEGRALES

- 1.1 INTRODUCTION
- 1.2 METHODE DES CONTRAINTES FICTIVES
- 1.3 METHODE DES DISCONTINUITES DES DEPLACEMENTS
- 1.4 METHODE DES EQUATIONS INTEGRALES DIRECTE
- 1.5 CHOIX DE LA METHODE
- 1.6 METHODE DES ELEMENTS FINIS ET METHODES DES EQUATIONS INTEGRALES
- 1.7 HISTORIQUE

CHAPITRE 2 :

FORMULATION DES EQUATIONS INTEGRALES EN TROIS DIMENSIONS

- 2.1 NOTATION INDICIELLE
- 2.2 HYPOTHESES
 - 2.2.1 HYPOTHESES SUR LE MILIEU
 - 2.2.2 HYPOTHESES SUR LES TYPES DE CHARGEMENTS
 - 2.2.3 HYPOTHESES SUR LA GEOMETRIE DU MILIEU
- 2.3 RAPPEL SUR L'ELASTICITE LINEAIRE
 - 2.3.1 TENSEUR DES CONTRAINTES
 - 2.3.2 EQUATIONS D'EQUILIBRE
 - 2.3.3 TENSEUR DES DEFORMATIONS
 - 2.3.4 LOI DE HOOKE GENERALISEE
 - 2.3.5 FORMULATION DU PROBLEME D'ELASTICITE
- 2.4 PROBLEME DE KELVIN
- 2.5 THEOREME DE RECIPROCITE DE MAXWELL-BETTI
- 2.6 CALCUL DES DEPLACEMENTS ET DES CONTRAINTES SUR LA SURFACE
- 2.7 CALCUL DES DEPLACEMENTS DES POINTS INTERIEURS
- 2.8 CALCUL DU TENSEUR DES CONTRAINTES DES POINTS INTERIEURS

CHAPITRE 3 :

TRAITEMENT NUMERIQUE DE L'EQUATION INTEGRALE

- 3.1 INTRODUCTION
- 3.2 DISCRETISATION DE L'EQUATION INTEGRALE
- 3.3 INTEGRATION NUMERIQUE
 - 3.3.1 CHOIX DE LA METHODE D'INTEGRATION
 - 3.3.2 METHODE DE QUADRATURE DE GAUSS
 - 3.3.3 CHOIX DU NOMBRE DE POINTS DE GAUSS
- 3.4 CONSTITUTION DU SYSTEME D'EQUATIONS
 - 3.4.1 PROBLEME DE DIRICHLET
 - 3.4.2 PROBLEME DE NEUMAN
 - 3.4.3 PROBLEME MIXTE
- 3.5 RESOLUTION DU SYSTEME D'EQUATION
 - 3.5.1 CONDITIONNEMENT DE MATRICE
 - 3.5.2 CHOIX DE LA METHODE DE RESOLUTION
- 3.6 DETERMINATION DES DEPLACEMENTS DES POINTS INTERIEURS
- 3.7 DETERMINATION DES TENSUEURS DES CONTRAINTES DES POINTS INTERIEURS

CHAPITRE 4 :

ORGANISATION ET CARACTERISTIQUES GENERALES DU PROGRAMME

- 4.1 CARACTERISTIQUES DU PROGRAMME
- 4.2 ORGANIGRAMME
- 4.3 SUBROUTINES DU PROGRAMME
- 4.4 SCHEMA REPRESENTANT LES LIAISONS ENTRE LES SUBROUTINES

CHAPITRE 5 :

EXEMPLES

- 5.1 DEPLACEMENTS DU CORPS RIGIDE
- 5.2 CUBE EN TRACTION

CONCLUSION

ANNEXE : LISTING DU PROGRAMME M.E.I.T.D

INTRODUCTION

INTRODUCTION

Sous la poussée des besoins nouveaux, issus du développement des techniques les plus récentes, on est amené à réaliser des projets de plus en plus complexes, tout en augmentant les performances et en réduisant les coûts : L'aérospatiale, le nucléaire, l'exploitation des océans, contrôle de la pollution, aménagement des cours d'eau , etc.

Pour pouvoir réaliser ces projets , on a besoin de modèles qui nous permettent de simuler le comportement de ces systèmes physiques complexes . Le comportement de plusieurs phénomènes physiques peut être décrit par des équations aux dérivées partielles à coefficients constants: mécanique des solides et des fluides, le transfert de chaleur et de masse, etc .

La méthode des équations intégrales est l'une des méthodes les plus puissantes qui nous permet de résoudre efficacement ces équations .

Elle est applicable dans la majorité de problèmes rencontrés dans la pratique : problèmes stationnaires ou non stationnaires, linéaires ou non linéaires, définis dans un domaine de géométrie quelconque, à une, deux ou trois dimensions .

Elle consiste à utiliser la solution analytique d'un problème particulier (problème de KELVIN en élastostatique), où le domaine est infini , et on applique une seule perturbation en un point de ce domaine. En superposant les solutions particulières on obtient la solution du problème posé .

Dans cette étude on se limite au cas élastostatique dans la mécanique des solides .

Le but est d'élaborer un logiciel qui permet la détermination du champ des déplacements et le champ des contraintes dans un corps homogène, élastique et linéaire , en trois dimensions .

Cette thèse est organisée en cinq chapitres :

Le premier chapitre présente les trois versions de la méthode des équations intégrales , fait une comparaison qualitative entre la méthode des équations intégrales et la méthode des éléments finis et parle de l'évolution de cette méthode .

Dans le second chapitre, on présente la formulation de la méthode des équations intégrales directe en trois dimensions.

Le chapitre trois expose le traitement numérique des équations intégrales.

Dans le chapitre quatre on présente l'organigramme et les caractéristiques générales du programme, et dans le chapitre cinq on traite quelques exemples pour mettre en évidence l'adaptation de cette méthode aux problèmes tridimensionnels.

Et enfin , le listing du programme est donné en annexe .

CHAPITRE 1

GENERALITES SUR LES METHODES DES EQUATIONS INTEGRALES

1.1 INTRODUCTION :

Tout phénomène physique régi par :

- des équations aux dérivées partielles linéaires à coefficients constants à l'intérieur d'une région R .
- et par des conditions aux limites prescrites sur la surface S de la région R .

peut être traité avantageusement par la méthode des équations intégrales.

Comme exemples de ces phénomènes physiques :

- Le transfert de masse.
- Le transfert de chaleur.
- L'écoulement des fluides.
- L'électrostatique.
- La mécanique des solides ... etc

Pour le transfert de chaleur en régime permanent l'équation différentielle qui le gouverne est l'équation de LAPLACE :

$$\nabla^2 T = 0$$

où T est la température.

En mécanique des solides les équations aux dérivées partielles sont les équations d'équilibre élastique , où les inconnues sont les contraintes , ou sous une autre forme , l'équation de NAVIER dont les inconnues sont les déplacements.

La solution analytique pour ce genre de problèmes n'existe que pour des cas très simples , où la région R est homogène , de géométrie très simple et avec des conditions aux limites relativement franches.

Comme exemple de ces cas simples : Une perturbation en un point (une force concentrée ou une discontinuité de déplacement en mécanique des solides , ou une source de chaleur en transfert de chaleur) , dans une région homogène , infinie.

Cette solution analytique est la base des méthodes des équations intégrales . Elle est appelée solution singulière car mathématiquement parlant , elle a un bon comportement en chaque point sauf là où la perturbation est appliquée , il y'a une anomalie mathématique ou singularité.

Pour un problème donné , la solution sera une combinaison linéaire des solutions singulières dont leurs effets sur le contour seront les conditions aux limites .

Historiquement , la méthode des équations intégrales a été développée de deux manières distinctes :

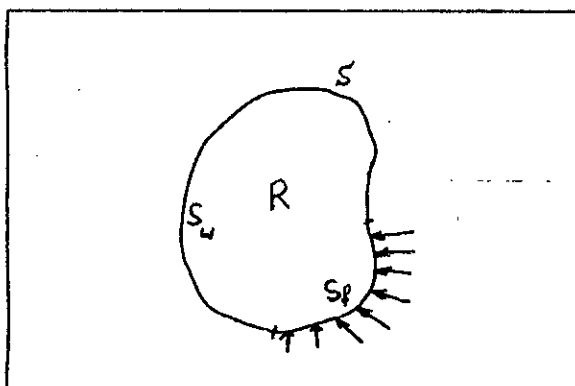
- L'une des deux est une approche physique intuitive :
 - * La méthode des contraintes fictives (ou méthode de discontinuités des contraintes) .
 - * La méthode des discontinuités des déplacements.

Dans cette approche on cherche en premier lieu les valeurs des perturbations fictives dont les effets sur la surface sont les conditions aux limites spécifiées , ensuite on calcule le reste des paramètres de la surface comme étant les effets de ces perturbations " fictives " en ces points .

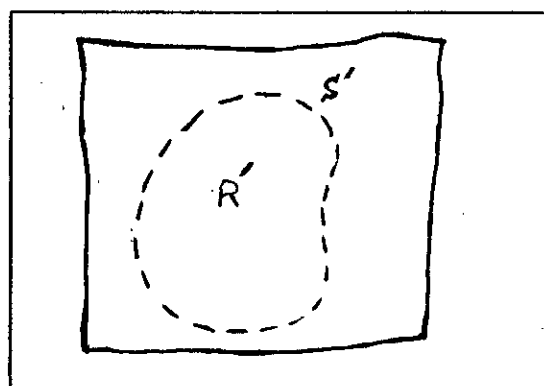
Puisque les paramètres de la surface sont obtenus indirectement, cette approche est appelée méthode des équations intégrales INDIRECTE.

L'autre approche est une approche mathématique , elle est basée sur certains théorèmes fondamentaux (théorème de réciprocité de MAXWELL-BETTI en mécanique des solides) qui éliminent les paramètres intermédiaires , et relie directement les inconnues de la surface aux conditions limites. Cette approche est appelée la méthode des équations intégrales DIRECTE .

1.2 METHODE DES CONTRAINTES FICTIVES :



a) Corps à étudier



b) Trace du corps à étudier dans un milieu infini

Fig 1.1

La figure (1.1.a) représente la région R limitée par la surface S du problème à résoudre .

la fig (1.1.b) représente une région infinie, et S' est la trace de la surface S dans cette région .

Si on applique une charge concentrée en un point dans la région infinie , on aura les déplacements et les contraintes en chaque point de la région infinie (c'est la solution analytique du problème de KELVIN) .

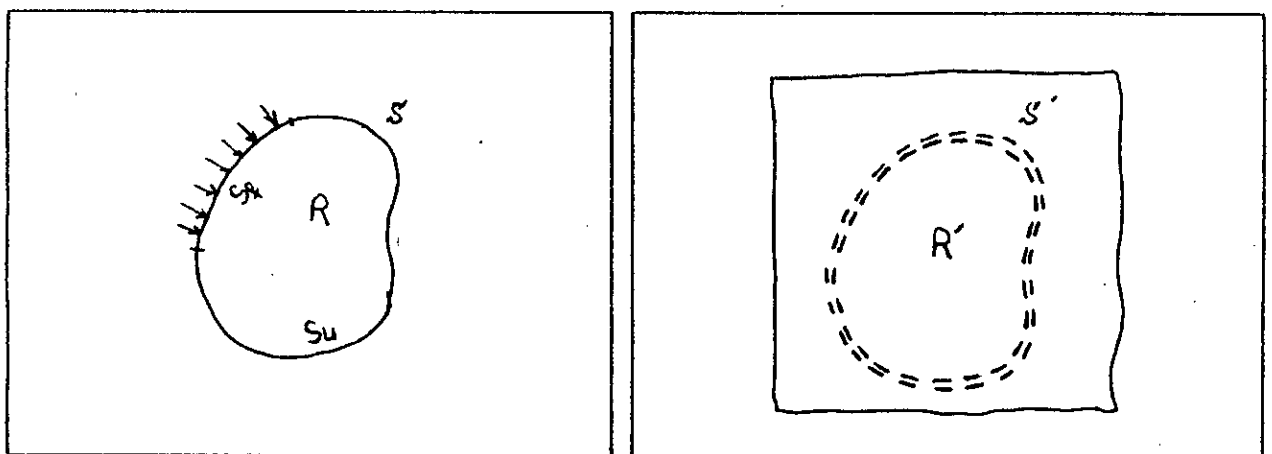
La méthode des contraintes fictives consiste à trouver des contraintes fictives qui produisent les mêmes conditions sur S' en milieu infini que ceux prescrites sur S , et puisque la solution du problème est unique alors on aura résolu le problème (a) en résolvant le problème (b) .

La démarche numérique est la suivante :

On discrétise S' en N éléments , alors on doit satisfaire $3N$ conditions aux limites . pour cela on cherche $3N$ contraintes fictives qu'on applique sur les N éléments (une contrainte pour chaque élément et chaque direction) , les effets de ces $3N$ contraintes sur S' doivent être les conditions aux limites sur S .

Donc on peut écrire un système de $3N$ équations algébriques linéaires à $3N$ inconnues . Une fois ces contraintes fictives obtenues , la solution en chaque point de R sera l'effet de ces contraintes fictives en ce points.

1.3 METHODE DES DISCONTINUITES DES DEPLACEMENTS :



a) Corps à étudier

b) Trace du corps à étudier dans un milieu infini

Fig 1.2

La figure (1.2.a) représente la région R limitée par la surface S.

La figure (1.2.b) représente une région infinie et S' est la surface dans cette région.

Plusieurs problèmes pratiques dans la mécanique des solides font intervenir des corps qui contiennent des fissures.

La méthode des contraintes fictives ne s'adapte pas pour ce genre de problèmes car les effets des contraintes fictives le long d'une surface de la fissure ne se distinguent pas de ceux sur l'autre surface. La méthode de discontinuité des déplacements peut cependant prendre en compte ce genre de problèmes. Elle est basée sur la solution analytique du problème d'une discontinuité de déplacement constante dans un corps élastique infini.

Physiquement on doit imaginer une discontinuité de déplacements comme une fissure représentée par deux surfaces qui ont un déplacement relatif entre elles.

Pour obtenir la solution numérique du problème considéré, on discrétise la surface fictif S' en N éléments et on cherche 3N discontinuités de déplacements, dont les effets sur S' sont les conditions aux limites sur la surface S. Cela revient à résoudre un système d'équation de 3N equations à 3N inconnues.

Une fois ces discontinuités de déplacements obtenues, la solution en chaque point de la région R sera la superposition des effets de ces discontinuités de déplacements en ce point.

1.4 METHODE DES EQUATIONS INTEGRALES DIRECTE :

Elle est basée sur le théorème de MAXWELL-BETTI qui fait le lien direct entre les paramètres de la surface S du problème qu'on veut résoudre $\left[u_j(Q), t_j(Q) \right]$ avec la solution analytique du problème de KELVIN $\left[U_{ij}(P,Q), T_{ij}(P,Q) \right]$:

$$\frac{1}{2} \delta_{ij} u_j(P) + \int_S T_{ij}(P,Q) u_j(Q) dS(Q) = \int_S U_{ij}(P,Q) t_j(Q) dS(Q)$$

Où :

S est la surface du corps .

P et Q sont des points de la surface S .

On discrétise la surface S en N éléments , et on suppose que u_j et t_j sont constants sur chaque éléments .

on aura ainsi 6N paramètres sur la surface (3N déplacements u_j $j=1,2,3$, et 3N tractions t_j $j=1,2,3$) .

Pour un problème bien posé , la moitié de ces paramètres sont donnés comme conditions aux limites , et on doit trouver les 3N paramètres restants .

En faisant varier P sur les N éléments et les trois directions \vec{x}_i , on aura 3N équations de la forme :

$$\frac{1}{2} \delta_{ij} u_j(P) + \sum \left\{ \int_{\Delta S(Q)} T_{ij}(P,Q) dS(Q) \right\} u_j(Q) = \sum \left\{ \int_{\Delta S(Q)} U_{ij}(P,Q) dS(Q) \right\} t_j(Q)$$

On a ainsi 3N équations algébriques à 3N inconnues .

Une fois le système algébrique résolu , \vec{u} et \vec{t} seront connus sur chaque point de la surface .

Les valeurs des déplacements et des contraintes pour les points intérieurs seront calculés par simple intégration , en utilisant l'équation de SOMIGLIANA :

$$u_i(P) = - \int_S T_{ij}(P,Q) u_j(Q) dS(Q) + \int_S U_{ij}(P,Q) t_j(Q) dS(Q)$$

et comme :

$$\sigma_{ij} = \lambda \varepsilon_{kk} \delta_{ij} + 2 \mu \varepsilon_{ij} \quad \text{et} \quad \varepsilon_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i})$$

On aura alors :

$$\sigma_{ij}(P) = - \int_S S_{ijk}(P,Q) u_k(Q) dS(Q) + \int_S D_{ijk}(P,Q) t_k(Q) dS(Q)$$

S_{ijk} et D_{ijk} sont des noyaux qui dérivent respectivement de T_{ij} et U_{ij} .

1.5 CHOIX DE LA METHODE :

Dans cette étude nous avons opté pour la méthode intégrale directe pour les raisons suivantes :

1- La méthode des discontinuités des déplacements a été écartée, car on n'étudie pas les problèmes de fissurations où elle est très efficace .

2- Du point de vue formulation, la méthode intégrale directe est plus puissante, car elle a une base mathématique, alors que la méthode des contraintes fictives a une base physique intuitive .

3- Du point de vue temps de calcul , la méthode des contraintes fictives prend plus de temps, car on cherche en premier lieu les contraintes fictives , ensuite on détermine les paramètres de la surface , alors qu'avec la méthode directe on les détermine directement .

4- L'utilisation des éléments d'ordre supérieur avec la méthode directe est très simple , alors qu'avec la méthode des contraintes fictives on trouve une grande difficulté à les utiliser

1.6 METHODE DES ELEMENTS FINIS ET METHODES INTEGRALES :

Le développement des méthodes numériques pour la résolution des problèmes de la mécanique , a connu ces trente dernières années un essor remarquable , en particulier la méthode des éléments finis, qui couvre un large domaine d'applications en calcul des structures : l'élasticité , la plasticité , la mécanique des fluides...etc.

Cette méthode présente une grande souplesse , mais son utilisation présente un certain nombre de problèmes :

1 - Le coût de préparation des données est très élevé et cela est dû à :

- a) - Le maillage s'étend sur tous le domaine.
- b) - Nécessité de respecter la connectivité entre les éléments.

2 - Elle nécessite la résolution d'un grand nombre d'équations simultanément , mais il faut noter que le système à résoudre est symétrique et peu peuplé.

3 - La plus part des informations données par la M.E.F ne sont pas utilisées .

4 - Le volume des résultats est très grand de telle façon qu'on trouve une difficulté pour la sélection des résultats.

5 - La M.E.F est difficile à mettre en oeuvre pour les problèmes tridimensionnels, les problèmes extérieurs (domaine infinie), et les problèmes de fissurations.

6 - La version déplacement de la M.E.F qui est la plus ordinairement utilisée donne de bon résultats pour les déplacements , alors que la précision est moindre pour les contraintes (discontinuité des contraintes à travers l'interface des éléments) qui sont dans la plus part des cas les plus intéressantes à connaître.

Ces difficultés associées à la M.E.F laissent l'utilisation de la méthode des équations intégrales plus commode dans beaucoup de problèmes :

La méthode des équations intégrales consiste à transformer les équations aux dérivées partielles qui décrivent le comportement d'une fonction inconnue à l'intérieur et aux frontières d'un domaine en des équations intégrales sur les frontières seulement.

Du point de vue analyse numérique , la dimension du problème se réduit de un , ce qui simplifie avantageusement le traitement numérique et spécialement on a :

- 1 - Le coût de préparation des données est faible:
 - a) - Le maillage se limite aux frontières seulement.
 - b) - La connectivité entre les éléments n'est pas exigée donc on peut raffiner le maillage sans aucune difficulté.
- 2 - L'interprétation des résultats est très simple car les résultats sont obtenus en des points que l'on précise préalablement.
- 3 - L'ordre du système des équations à résoudre simultanément est énormément réduit, mais il faut noter que la matrice est non symétrique et très peuplée.
- 4 - La M.E.I est très efficace pour les problèmes extérieurs (domaines infinis), les problèmes tridimensionnels ainsi que les problèmes de fissurations.
- 5 - En fin, parce qu'elle exploite une solution analytique, la méthode des équations intégrales est potentiellement plus précise que la méthode des éléments finis, où une solution approximative est faite en chaque élément de la région R.

Et voici quelques exemples à titre indicatif qui ont été traités par le C.E.T.I.M, par les deux méthodes numériques :

1 - CYLINDRE EPAIS :

Il s'agit d'un secteur de 45° d'un cylindre creux pour lequel on fait varier l'épaisseur, et on compare le temps d'exécution pour une même précision sur les contraintes. Ils ont abouti aux résultats résumés dans la figure (fig 1.3.b).

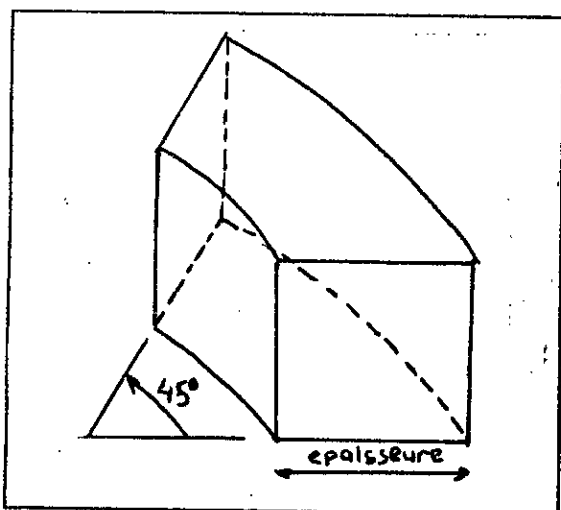


fig 1.3.a
Secteur de 45° d'un cylindre creux.

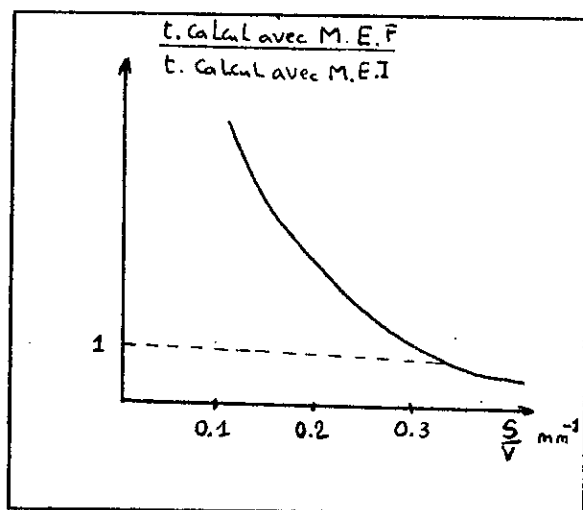


fig 1.3.b
Rapport des temps de calcul M.E.F et M.E.I en fonction du rapport (surface / volume) de la pièce à précision égale de contraintes.

fig 1.3

2 - ETUDE D'UNE EPROUVETTE CT15 :

Pour aboutir à un même résultat il a fallu :

- En M.E.F : 223 éléments.
1010 degrés de liberté.
- En M.E.I : 28 éléments 168 degrés de liberté.

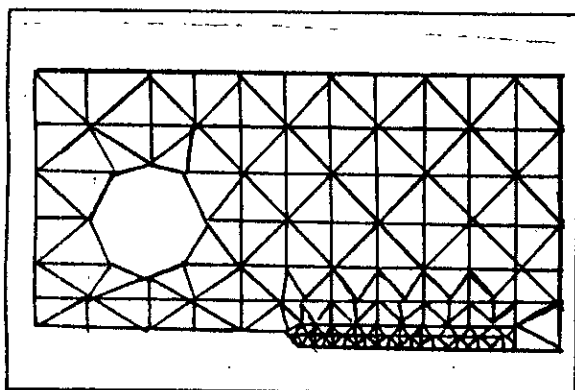


fig 1.4.a
Maillage de l'éprouvette avec la méthode des éléments finis

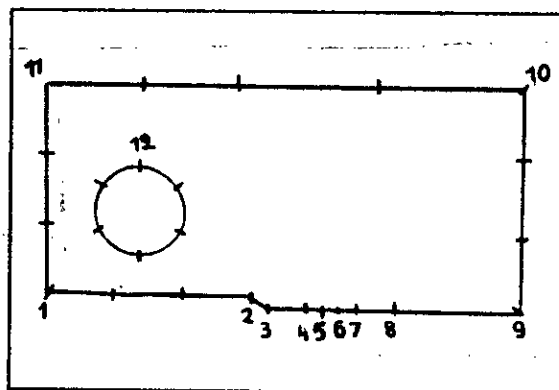


fig 1.4.b
Maillage de l'éprouvette avec la méthode des équations intégrales

fig 1.4

1.7 HISTORIQUE :

Du point de vue historique , les méthodes intégrales sont le résultat du développement de la méthode de FREDHOLM pour la résolution des problèmes de potentiel régis par l'équation de LAPLACE :

- Diffusion de la chaleur en régime permanent.
- Ecoulements à potentièl.
- L'électrostatique ...

FREDHOLM est le premier qui a appliqué la méthode intégrale à l'élasticité en 1905 , et cela pour résoudre des problèmes assez simples regroupés sous le nom du premier problème fondamental (les conditions aux limites sont les déplacements).

Il trouve que le système des équations aux dérivées partielles de NAVIER est un système élliptique , et le problème d'intégration quand les déplacements sur les frontières sont connus ,est un problème de DIRICHLET, donc la solution est analogue à la solution du problème de DIRICHLET pour les fonctions harmoniques.

Jusqu'en 1950 , le second problème fondamental (les conditions aux limites sont les contraintes) ,ainsi que le troisième problème fondamental (conditions aux limites mixtes) restent non résolubles avec les équations de FREDHOLM.

C'est en 1956 que KINOSHITA et MURA ont obtenu par la méthode du potentiel , des équations intégrales pour les deux premiers problèmes fondamentaux. En 1963 KUPRADZE a reconsidéré le problème pour un cas plus général .il a établi une démonstration générale et rigoureuse basée sur la théorie des équations intégrales singulières et multidimensionnelles , et présenté deux méthodes numériques approximatives de l'équation integrale.

En 1967 RIZZO a analysé par ordinateur quelques exemples simples de l'élastostatique , et en 1968 CRUSE a généralisé la méthode à l'élastodynamique.

En 1971 SWEDLOW et CRUSE ont présenté une analyse élastoplastique pour les matériaux anisotrope compressible. Les travaux de CRUSE ont montré qu'on peut modéliser des formes de géométries complexes avec des conditions aux limites non uniformes.

En 1975 C.LACHAT, dans sa thèse de Ph.D , a développé la méthode intégrale pour les problèmes d'élastostatique en deux et trois dimensions en utilisant les éléments isoparamétriques avec des variations linéaires , quadratiques ou cubiques des variables.

CHAPITRE 2

FORMULATION DES EQUATIONS INTEGRALES EN TROIS DIMENSIONS

2.1 NOTATION INDICIELLE :

Deux conventions sont utilisées pour la notation indicielle :

- La première : si l'indice est répété alors il y a sommation.

exemple :

$$\sigma_{ij} n_j = \sum_{j=1}^3 \sigma_{ij} n_j = \sigma_{i1} n_1 + \sigma_{i2} n_2 + \sigma_{i3} n_3 \quad (2.1)$$

- La seconde : un indice précédé par une virgule indique qu'il y a une différentiation partielle avec la variable représentée par cet indice .

exemple :

$$u_{i,jk} = \frac{\partial u_i^2}{\partial x_j \partial x_k} \quad (2.2)$$

En combinant ces deux conventions on aura par exemples :

$$u_{i,jj} = \frac{\partial u_i^2}{\partial x_1^2} + \frac{\partial u_i^2}{\partial x_2^2} + \frac{\partial u_i^2}{\partial x_3^2} \quad (2.3)$$

2.2 HYPOTHESES :

2.2.1 HYPOTHESES SUR LE MILIEU :

On considère un milieu :

- CONTINU : Le milieu ne contient pas de fissure ou de discontinuité .

- ELASTIQUE : Les déformations sont rattrapées lorsque les contraintes sont supprimées .

- LINEAIRE : Chaque composante du tenseur des contraintes est exprimée comme une combinaison linéaire de toutes les composantes du tenseur des déformations .

- HOMOGENE , ISOTROPE : Les composantes reliant le tenseur des contraintes au tenseur des déformations , qui caractérisent la propriété d'élasticité du milieu considéré , sont indépendantes de la position (homogénéité), et indépendantes de l'orientation (isotropie) .

2.2.2 HYPOTHESES SUR LES TYPES DE CHARGEMENTS :

- On ne considère pas les contraintes d'origine thermique.
- On ne considère que les forces surfaciques : les forces de volume sont négligées .
- Les chargements réels sont des forces réparties , une force concentrée sera considérée comme étant une force répartie sur un élément de très petites dimensions .
- On considère que les chargements sont quasi-statiques.

2.2.3 HYPOTHESES SUR LA GEOMETRIE DU MILIEU :

La surface chargée de la région à étudier doit être limitée :

- Pour les problèmes intérieurs , le milieu a un volume fini donc une surface finie . Ce type de problème est résolvable avec cette méthode .

- Pour les problèmes extérieurs : le milieu a un volume infini , il faut que la surface soit finie : comme exemple : un trou sphérique dans un milieu infini soumis à une pression constante .(fig 2.1.a)

Cette hypothèse exclut tous les problèmes où la surface sera infinie , exemples :

Un chargement sur un plan infini .(fig 2.1.b)

Un trou cylindrique débouchant , dans un milieu infini, soumis à une pression constante .(fig 2.1.c)

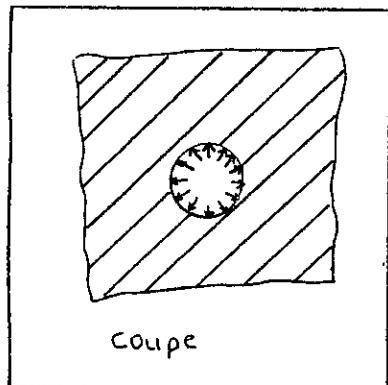


fig (2.1.a)
trou sphérique dans
un milieu infini

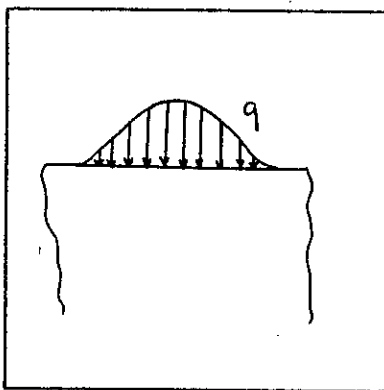


fig (2.1.b)
Chargement sur un
plan infini

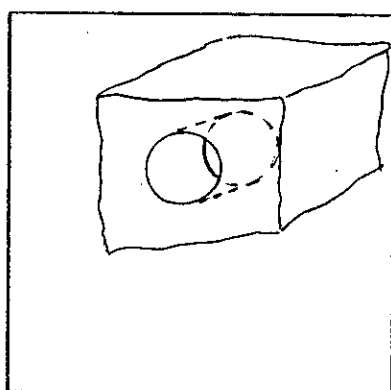


fig (2.1.c)
Trou débouchant

fig 2.1

Ce type de problèmes peut être résolu avec la méthode des équations intégrales en utilisant la solution analytique du problème de FLAMANT (une force concentrée sur un plan infini.), et non pas avec la méthode des équations intégrales directe qui utilise la solution analytique du problème de KELVIN (une force concentrée dans un milieu infini) qui est utilisée dans notre étude .

2.3 RAPPEL SUR L'ELASTICITE LINEAIRE :

2.3.1 - TENSEUR DES CONTRAINTES :

Les contraintes agissantes sur les plans dont leurs normales sont parallèles aux axes sont appelées les composantes du tenseur des contraintes.

Les composantes du tenseur des contraintes sont notées par

σ_{ij} :

i : indique la direction de la normale du plan où la contrainte est exercée .

j : indique la direction de la composante de la contrainte.

2.3.2 - EQUATIONS D'EQUILIBRE :

La condition pour que le corps soit en équilibre impose certaines conditions sur les composantes du tenseur des contraintes :

- La première , trouvée à partir de l'équilibre des moment, est que le tenseur des contraintes soit symétrique .

$$\sigma_{ij} = \sigma_{ji} \quad (i, j = 1, 2, 3) \quad (2.4)$$

- La seconde , trouvée à partir de l'équilibre des forces, est que les composantes du tenseur des contraintes doivent satisfaire les équations suivantes :

$$\left\{ \begin{array}{l} \frac{\partial \sigma_{11}}{\partial x_1} + \frac{\partial \sigma_{21}}{\partial x_2} + \frac{\partial \sigma_{31}}{\partial x_3} + F_1 = 0 \\ \frac{\partial \sigma_{12}}{\partial x_1} + \frac{\partial \sigma_{22}}{\partial x_2} + \frac{\partial \sigma_{32}}{\partial x_3} + F_2 = 0 \\ \frac{\partial \sigma_{13}}{\partial x_1} + \frac{\partial \sigma_{23}}{\partial x_2} + \frac{\partial \sigma_{33}}{\partial x_3} + F_3 = 0 \end{array} \right. \quad (2.5)$$

où F_1, F_2, F_3 sont les forces de volume.

Ou d'une façon plus contractée : $\sigma_{ij,j} + F_i = 0$

Les équations d'équilibres donnent trois équations pour les six inconnues :

$$(\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12} = \sigma_{21}, \sigma_{13} = \sigma_{31}, \sigma_{23} = \sigma_{32}).$$

Ainsi, les équations d'équilibres ne suffisent pas pour calculer les contraintes, trois autres équations sont nécessaires.

2.3.3 TENSEUR DES DEFORMATIONS :

L'état de déformation en un point dans un corps est spécifié par les composantes du tenseur des déformations.

Elles sont définies en terme de déplacement u_i ($i=1,2,3$) :

$$\varepsilon_{ij} = \frac{1}{2} \left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right] \quad (i,j=1,2,3) \quad (2.6)$$

On a, par définition : $\varepsilon_{ij} = \varepsilon_{ji}$

2.3.4 LOI DE HOOKE GENERALISEE :

La théorie classique de l'élasticité linéaire isotrope est basée sur la loi généralisée de HOOKE : à chaque point d'un corps élastique linéaire les six composantes du tenseur des contraintes ($\sigma_{ij} = \sigma_{ji}$) sont linéairement reliées aux six composantes du tenseur de déformation ($\varepsilon_{ij} = \varepsilon_{ji}$).

Les constantes reliant les composantes du tenseur des contraintes aux composantes du tenseur des déformations caractérisent la propriété d'élasticité du corps considéré .

dans notre étude on suppose que cette propriété est indépendante de la position et de l'orientation. (ie . le corps est homogène et isotrope).

Les relations DEFORMATIONS - CONTRAINTES s'écrivent :

$$\left\{ \begin{array}{l} \varepsilon_{11} = \frac{1}{E} [\sigma_{11} - \nu (\sigma_{22} + \sigma_{33})] \\ \varepsilon_{22} = \frac{1}{E} [\sigma_{22} - \nu (\sigma_{11} + \sigma_{33})] \\ \varepsilon_{33} = \frac{1}{E} [\sigma_{33} - \nu (\sigma_{11} + \sigma_{22})] \\ \varepsilon_{12} = \frac{1}{2G} \sigma_{12} \\ \varepsilon_{13} = \frac{1}{2G} \sigma_{13} \\ \varepsilon_{23} = \frac{1}{2G} \sigma_{23} \end{array} \right. \quad (2.7.1)$$

$$\text{Soit : } \varepsilon_{ij} = \frac{1+\nu}{E} \sigma_{ij} - \frac{\nu}{E} \sigma_{kk} \delta_{ij} \quad (2.7.2)$$

où :

E : module de YOUNG

ν : coefficient de poisson

G : module de cisaillement

Il sont reliés par la relation suivante :

$$G = \frac{1}{2} \frac{E}{(1+\nu)} \quad (2.8)$$

Les relations CONTRAINTES - DEFORMATIONS s'écrivent :

$$\left\{ \begin{array}{l} \sigma_{11} = \frac{2G}{(1-2\nu)} \left[(1-\nu) \varepsilon_{11} + \nu (\varepsilon_{22} + \varepsilon_{33}) \right] \\ \sigma_{22} = \frac{2G}{(1-2\nu)} \left[(1-\nu) \varepsilon_{22} + \nu (\varepsilon_{11} + \varepsilon_{33}) \right] \\ \sigma_{33} = \frac{2G}{(1-2\nu)} \left[(1-\nu) \varepsilon_{33} + \nu (\varepsilon_{11} + \varepsilon_{22}) \right] \\ \sigma_{12} = 2G \varepsilon_{12} \\ \sigma_{13} = 2G \varepsilon_{13} \\ \sigma_{23} = 2G \varepsilon_{23} \end{array} \right. \quad (2.9.1)$$

ou sous une forme plus contractée :

$$\sigma_{ij} = \lambda \varepsilon_{kk} \delta_{ij} + 2\mu \varepsilon_{ij} \quad (2.9.2)$$

où λ et μ sont les coefficients de LAME :

$$\lambda = 2\nu G / (1-2\nu) \quad \text{et} \quad \mu = E / 2(1+\nu) = G \quad (2.10)$$

2.3.5 FORMULATION DU PROBLEME D'ELASTICITE :

L'objet de la résolution du problème d'élasticité est de trouver le champ des déplacements ainsi que le champ des contraintes pour un corps élastique qui est soumis à des forces volumiques et à des forces surfaciques sur une partie de la surface et on lui impose des déplacements sur l'autre partie de la surface .

Cela veut dire trouver les six composantes de contraintes

$(\sigma_{ij} = \sigma_{ji})$, et les trois composantes du vecteur déplacement u_i .

Les trois équations d'équilibres et les six relations DEFORMATIONS-CONTRAINTES donnent neuf équations pour ces neuf inconnues

$$\left[\begin{array}{l} \sigma_{ij,j} + \beta_i = 0 \quad (i=1,2,3) \\ \sigma_{ij} = 2G \left(\varepsilon_{ij} + \frac{\nu}{(1+2\nu)} \varepsilon_{kk} \delta_{ij} \right) \quad (i,j=1,2,3) \end{array} \right. \quad (2.11)$$

Si on suppose que les déplacements u_i sont des fonctions continues, ces neuf équations sont suffisantes pour déterminer les neuf quantités inconnues .

Les conditions aux limites font la distinction entre un problème élastique et un autre , elles servent uniquement pour définir un problème particulier qu'on résoud .

Si on exprime les équations d'équilibre en fonction des déplacements , alors on aura un système de trois équations à trois inconnues appelé équations de NAVIER:

$$(\lambda + \mu) u_{j,j} + \mu u_{i,jj} = - F_i \quad (i = 1,2,3) \quad (2.12)$$

ou sous une forme vectorielle :

$$(\lambda + \mu) \overrightarrow{\text{grad}} \text{div } \vec{u} + \mu \Delta \vec{u} = - \vec{F} \quad (2.13)$$

2.4 PROBLEME DE KELVIN :

Le problème de KELVIN est illustré sur la figure (2.2).

On applique une force concentrée F_i de valeur égale à l'unité , en un point 0 : origine du repère, dans un milieu infini , suivant une direction \vec{x}_i , Et on cherche le vecteur déplacement ainsi que le tenseur des contraintes en un point Q (Q_1, Q_2, Q_3).

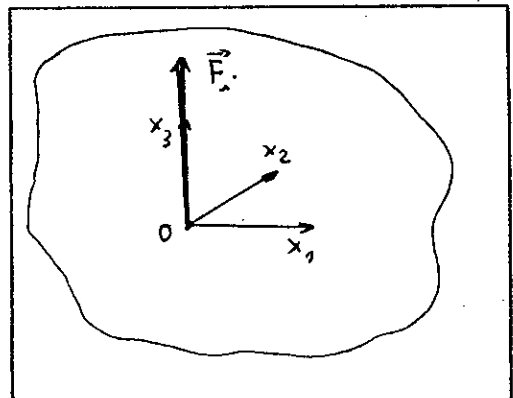


fig 2.2

Force concentrée dans un milieu infini.

soit $U_{ij}(0,Q)$ la composante du vecteur déplacement en Q suivant la direction x_j .

pour ce problème l'équation de NAVIER s'écrit :

$$(\lambda + \mu) u_{p,pj} + \mu u_{j,pp} = - \delta_{ij}(0, Q) \quad (2.14)$$

Et on a comme conditions aux limites : les déplacements à l'infini nuls.

$$\lim_{Q \rightarrow \infty} U_{ij}(0, Q) = 0 \quad (2.15)$$

La résolution de cette équation différentielle peut être effectuée par plusieurs méthodes mathématiques :

- Méthode de superposition .
- Transformation de FOURIER .
- Décomposition de $\delta(0, Q)$ sous formes d'ondes planes
- La méthode de HORMANDER .

La méthode de HORMANDER est la meilleur à utiliser car elle donne la solution analytique de toutes les équations différentielles élliptiques à coefficients constants .

On utilisant cette méthode on aura :

$$U_{ij}(0, Q) = \frac{(1+\nu)}{8 \pi E (1-\nu) r} \left\{ (3-4\nu)\delta_{ij} + \frac{Q_i Q_j}{r^2} \right\} \quad (2.16)$$

$$\text{Où} : r = \sqrt{Q_i Q_i} \quad (2.17)$$

maintenant si on applique la force concentrée en un point P (P_1, P_2, P_3) et en effectuant un changement d'origine on aura le déplacement d'un point Q suivant la direction x_j due à une force unité appliquée en P suivant la direction x_i :

$$U_{ij}(P, Q) = \frac{(1+\nu)}{8 \pi E (1-\nu) r} \left\{ (3-4\nu)\delta_{ij} + \frac{(P_i - Q_i)(P_j - Q_j)}{r^2} \right\} \quad (2.18)$$

$$\text{Où} : r = \sqrt{(P_i - Q_i)(P_i - Q_i)} \quad (2.19)$$

On vérifie bien que les conditions aux limites sont respectées

$$\lim_{Q \rightarrow \infty} U_{ij}(O, Q) = \lim_{r \rightarrow \infty} \frac{1}{r} = 0 \quad (2.20)$$

Le tenseur des contraintes en Q sera donné par :

$$\sigma_{jk}^l = \lambda \delta_{jk} u_{l,l} + \mu (u_{j,k} + u_{k,j}) \quad (2.21)$$

La composante du vecteur traction en un point Q sur un plan de normale $\vec{n}(Q) (n_1(Q), n_2(Q), n_3(Q))$, suivant la direction x_j sera :

$$T_{ij}(P, Q) = \sigma_{jk}^l n_k(Q) \quad (2.22)$$

On obtient ainsi :

$$T_{ij}(P, Q) = \frac{1}{8 \pi (1-\nu) r^2} \left\{ (1-2\nu) \left[n_i(Q) \frac{(P_i - Q_i)}{r} - n_j(Q) \frac{(P_j - Q_j)}{r} \right] + n_m(Q) \frac{(P_m - Q_m)}{r} \left[(1-2\nu) \delta_{ij} + \frac{(P_i - Q_i)(P_j - Q_j)}{r^2} \right] \right\} \quad (2.23)$$

2.5 THEOREME DE RECIPROCITE DE MAXWELL-BETTI :

Le théorème de réciprocité fait le lien entre les solutions de deux problèmes différents de valeurs aux limites pour une même région R .

C'est une conséquence directe de la linéarité des équations d'équilibre et de la loi généralisé de HOOKE .

Supposons que le premier problème admet pour solution le champ des déplacements $u_i(Q) (i=1,2,3)$, et pour composantes des vecteurs tractions $t_i(Q) (i=1,2,3)$, sur la surface S de la région R .

Supposons aussi qu'un second problème est caractérisé par le champ des déplacements $u_i'(Q)$ ($i=1,2,3$), et les composantes des vecteurs tractions $t_i'(Q)$ ($i=1,2,3$) sur la même surface S de la même région R .

Alors le théorème de réciprocité stipule que :

Le travail fait par le premier groupe de contraintes ($t_i(Q)$ $i=1,2,3$) en se déplaçant à travers le second groupe des déplacements ($u_i'(Q)$ $i=1,2,3$) est égale au travail fait par le second groupe des contraintes ($t_i'(Q)$ $i=1,2,3$) en se déplaçant à travers le premier groupe des déplacements ($u_i(Q)$ $i=1,2,3$).

Mathématiquement on a :

$$\int_S t_i'(Q) u_i(Q) dS(Q) = \int_S t_i(Q) u_i'(Q) dS(Q) \quad (2.24)$$

2.6 CALCUL DES DEPLACEMENTS ET DES CONTRAINTES SUR LA SURFACE

La fig (2.3.a) représente la région R limitée par la surface S , du problème considéré.

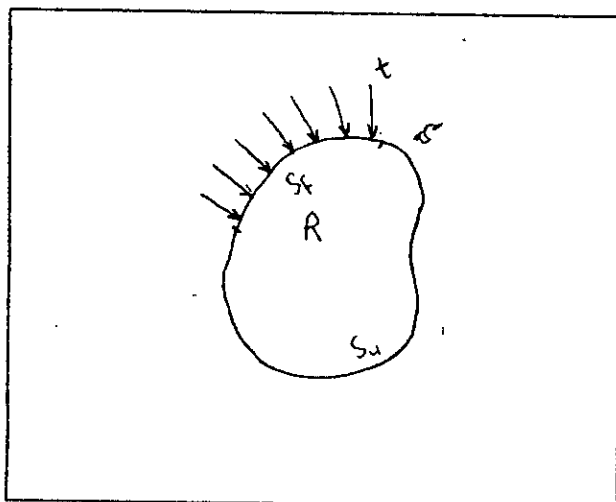


fig 2.3.a
Corps à étudier

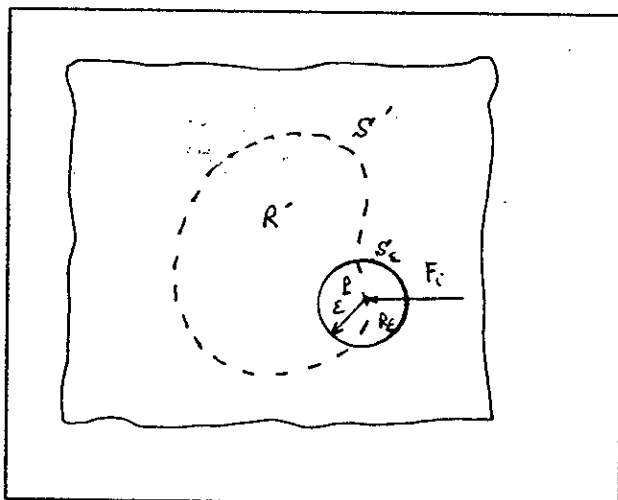


fig2.3.b
Trace du corps à étudier dans un milieu infini

fig 2.3

$u_j(Q)$: Le déplacement d'un point Q de la surface dans la direction x_j .

$t_j(Q)$: La composante du vecteur traction sur le plan tangent à la surface S en Q suivant la direction x_j .

Pour un problème bien posé, la moitié des paramètres $u_j(Q), t_j(Q)$, Q variant sur la surface, et $j = 1, 2, 3$, sont donnés comme conditions aux limites, l'autre moitié représente les inconnus qu'on doit retrouver.

La fig (2.3.b) représente une région infinie, et S' la trace de la surface S dans cette région (elle n'a pas de sens physique).

Si on applique une force concentrée égale à l'unité, en un point P du milieu infini, dans une direction x_i , alors le déplacement d'un point Q suivant la direction x_j est donné par la solution analytique du problème de KELVIN $U_{ij}(P, Q)$, et la composante du vecteur traction sur le plan tangent à S' en Q , suivant la direction x_j est donné par $T_{ij}(P, Q)$.

Si le point P est à l'extérieur de la région R' , alors R' sera en équilibre sous l'action du champ des contraintes $T_{ij}(P, Q)$.

Donc on a deux états où la région R' est en équilibre :

Le premier est le problème à résoudre $\left[u_j(Q), t_j(Q) \right]$

Le second est $\left[U_{ij}(P, Q), T_{ij}(P, Q) \right]$

Alors d'après le théorème de réciprocité de MAXWELL-BETTI on a :

$$\int_S T_{ij}(P, Q) u_j(Q) dS(Q) = \int_S U_{ij}(P, Q) t_j(Q) dS(Q) \quad (2.25)$$

Maintenant si on applique une force concentrée de valeur égale à l'unité, en un point P de la surface S' , dans une direction x_i , La région R' ne sera plus en équilibre sous l'action du champ des contraintes $T_{ij}(P, Q)$: il faut ajouter la force concentrée. Donc l'équation intégrale (2.25) ne peut pas être écrite sur la surface S' .

Pour pouvoir écrire l'équation précédente, on doit isoler le point P du contour où la force concentrée est appliquée, par une sphère de rayon ε .

La nouvelle surface où l'on intègre sera alors :

$$S - (R_\varepsilon \cap S) + (S_\varepsilon \cap R)$$

R_ε : Le volume de la sphère de rayon ε

S_ε : La surface de la sphère

Et on peut écrire :

$$\int_{S - (R_\varepsilon \cap S) + (S_\varepsilon \cap R)} T_{ij}(P, Q) u_j(Q) dS(Q) = \int_{S - (R_\varepsilon \cap S) + (S_\varepsilon \cap R)} U_{ij}(P, Q) t_j(Q) dS(Q) \quad (2.26)$$

En faisant tendre ε vers zero on

$$\lim_{\varepsilon \rightarrow 0} \int_{S - (R_\varepsilon \cap S) + (S_\varepsilon \cap R)} T_{ij}(P, Q) u_j(Q) dS(Q) =$$

$$\lim_{\varepsilon \rightarrow 0} \int_{S - (R_\varepsilon \cap S) + (S_\varepsilon \cap R)} U_{ij}(P, Q) t_j(Q) dS(Q) \quad (2.27)$$

La partie gauche de l'équation (2.27) est égale à :

$$\begin{aligned} & \lim_{\varepsilon \rightarrow 0} \int_{S - (R_\varepsilon \cap S)} T_{ij}(P, Q) u_j(Q) dS(Q) + \lim_{\varepsilon \rightarrow 0} \int_{(S_\varepsilon \cap R)} T_{ij}(P, Q) u_j(Q) dS(Q) \\ &= \int_S T_{ij}(P, Q) u_j(Q) dS(Q) + \lim_{\varepsilon \rightarrow 0} \int_{(S_\varepsilon \cap R)} T_{ij}(P, Q) u_j(Q) dS(Q) \quad (2.28) \end{aligned}$$

Calculons cette limite :

$$\lim_{\varepsilon \rightarrow 0} \int_{(S_\varepsilon \cap R)} T_{ij}(P, Q) u_j(Q) dS(Q) =$$

$$\lim_{\varepsilon \rightarrow 0} \int_{(S_\varepsilon \cap R)} \left\{ T_{ij}(P, Q) \left[u_j(Q) - u_j(P) \right] + T_{ij}(P, Q) u_j(P) \right\} dS(Q)$$

$$= \lim_{\varepsilon \rightarrow 0} \int_{(S_\varepsilon \cap R)} T_{ij}(P, Q) \left[u_j(Q) - u_j(P) \right] dS(Q) + \lim_{\varepsilon \rightarrow 0} \int_{(S_\varepsilon \cap R)} T_{ij}(P, Q) u_j(P) dS(Q)$$

On a :

$$\lim_{\varepsilon \rightarrow 0} \int_{(S_\varepsilon \cap R)} T_{ij}(P, Q) \left[u_j(Q) - u_j(P) \right] dS(Q) = 0 \quad (2.29)$$

en effet :

$$\left| T_{ij}(P, Q) \right| \leq \frac{k_1}{\varepsilon^2}$$

$$\left| u_j(Q) - u_j(P) \right| \leq k_2 \varepsilon \quad (\text{car } u_j \text{ est continue en chaque point})$$

$$(S_\varepsilon \cap R) \leq S_\varepsilon \leq 4 \pi \varepsilon^2$$

$$\text{donc } \lim_{\varepsilon \rightarrow 0} \int_{(S_\varepsilon \cap R)} T_{ij}(P, Q) \left[u_j(Q) - u_j(P) \right] dS(Q) \leq \lim_{\varepsilon \rightarrow 0} 8 \pi k_1 k_2 \varepsilon = 0$$

et :

$$\lim_{\varepsilon \rightarrow 0} \int_{(S_\varepsilon \cap R)} T_{ij}(P, Q) u_j(P) dS(Q) = \left\{ \lim_{\varepsilon \rightarrow 0} \int_{(S_\varepsilon \cap R)} T_{ij}(P, Q) dS(Q) \right\} u_j(P)$$

$$\text{Posons : } C_{ij}(P) = \lim_{\varepsilon \rightarrow 0} \int_{(S_\varepsilon \cap R)} T_{ij}(P, Q) dS(Q)$$

ainsi la partie gauche de l'équation (2.27) est égale à :

$$C_{ij}(P) u_j(P) + \int_S T_{ij}(P, Q) u_j(Q) dS(Q) \quad (2.30)$$

calculon la partie droite de l'équation (2.27) :

$$\lim_{\varepsilon \rightarrow 0} \int_{S - (R_\varepsilon \cap S) + (S_\varepsilon \cap R)} U_{ij}(P, Q) t_j(Q) dS(Q) =$$

$$\lim_{\varepsilon \rightarrow 0} \int_{S - (R_\varepsilon \cap S)} U_{ij}(P, Q) t_j(Q) dS(Q) + \lim_{\varepsilon \rightarrow 0} \int_{(S_\varepsilon \cap R)} U_{ij}(P, Q) t_j(Q) dS(Q)$$

$$= \int_S U_{ij}(P, Q) t_j(Q) dS(Q) + \lim_{\varepsilon \rightarrow 0} \int_{(S_\varepsilon \cap R)} U_{ij}(P, Q) t_j(Q) dS(Q) \quad (2.31)$$

la limite :

$$\lim_{\varepsilon \rightarrow 0} \int_{(S_\varepsilon \cap R)} U_{ij}(P, Q) t_j(Q) dS(Q) = 0 \quad (2.32)$$

en effet

$$\left| U_{ij}(P, Q) \right| \leq \frac{k_1}{\varepsilon}$$

$$\left| t_j(Q) \right| \leq k_2$$

$$\left| S_\varepsilon \cap R \right| \leq S_\varepsilon \leq 4 \pi \varepsilon^2$$

Donc
$$\lim_{\varepsilon \rightarrow 0} \int_{(S_\varepsilon \cap R)} U_{ij}(P, Q) t_j(Q) dS(Q) \leq \lim_{\varepsilon \rightarrow 0} 8 \pi k_1 k_2 \varepsilon = 0$$

la partie droite de l'équation (1) sera égale à :

$$\int_S U_{ij}(P, Q) t_j(Q) dS(Q) \quad (2.33)$$

finalement on a :

$$C_{ij}(P) u_j(P) + \int_S T_{ij}(P, Q) u_j(Q) dS(Q) = \int_S U_{ij}(P, Q) t_j(Q) dS(Q)$$

où :
$$C_{ij}(P) = \lim_{\varepsilon \rightarrow 0} \int_{(S_\varepsilon \cap R)} T_{ij}(P, Q) dS(Q) \quad (2.34)$$

Dans le cas où le vecteur normale à S en P est continue (ce qui est le cas de notre étude) on aura :

$$C_{ij}(P) = \frac{1}{2} \delta_{ij} \lim_{\varepsilon \rightarrow 0} \int_{S_\varepsilon} T_{ij}(P, Q) dS(Q) = \frac{1}{2} \delta_{ij}$$

on aura alors l'équation intégrales :

$$\frac{1}{2} \delta_{ij} u_j(P) + \int_S T_{ij}(P, Q) u_j(Q) dS(Q) = \int_S U_{ij}(P, Q) t_j(Q) dS(Q) \quad (2.35)$$

En résolvant cette équation intégrale u_j et t_j seront connus en chaque point de la surface.

2.7 CALCUL DES DEPLACEMENTS DES POINTS INTERIEURS :

Pour obtenir les déplacements pour les points intérieurs on suit une démarche analogue .

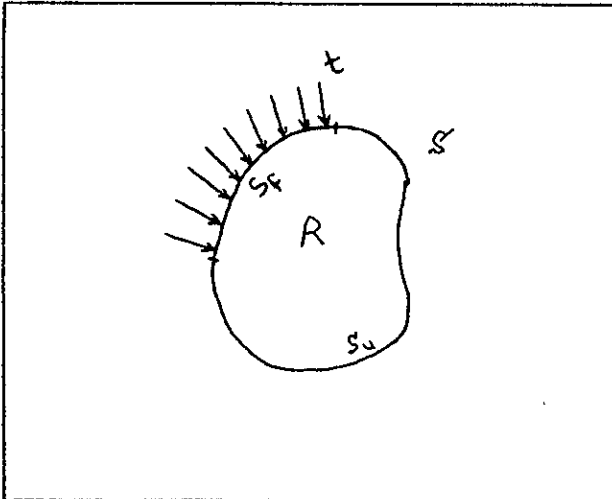


fig 2.4.a
Corps à étudier

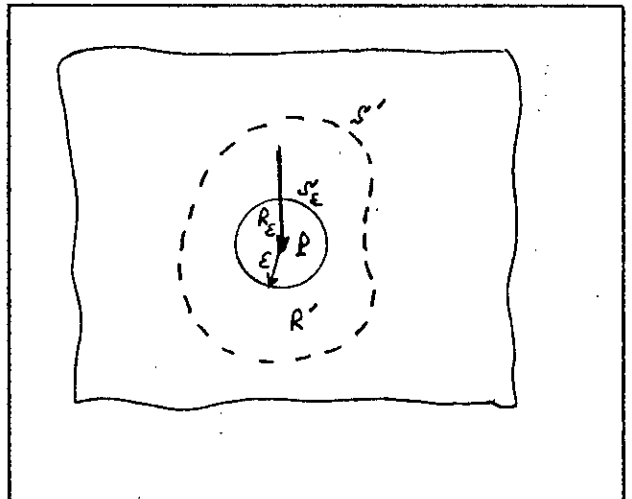


fig 2.4.b
Trace du corps à étudier
dans un milieu infini

fig 2.4

soit un point \$P\$ à l'intérieur de la région \$R'\$, où l'on veut calculer le vecteur déplacement et le tenseur des contraintes .

Appliquons en \$P\$ une force concentrée, qui a une valeur égale à l'unité, dans une direction \$x_j\$. Alors la solution analytique du problème de KELVIN nous donne le champ des tensions \$T_{ij}(P,Q)\$ sur la surface fictive \$S'\$, la région \$R'\$ ne sera pas en équilibre sous l'effet du champ des tensions \$T_{ij}(P,Q)\$ seulement, mais il faut leurs ajouter la force en \$P\$.

Pour pouvoir écrire l'équation de réciprocity de MAXWELL-BETTI , on isole le point \$P\$ par une sphère de rayon \$\epsilon\$.voir fig(2.4)

on aura alors :

$$\int_{S+S_\epsilon} T_{ij}(P,Q) u_j(Q) dS(Q) = \int_{S+S_\epsilon} U_{ij}(P,Q) t_j(Q) dS(Q) \quad (2.36)$$

en faisant tendre ε vers zero on aura :

$$\lim_{\varepsilon \rightarrow 0} \int_{S + S_\varepsilon} T_{ij}(P, Q) u_j(Q) dS(Q) =$$

$$\lim_{\varepsilon \rightarrow 0} \int_{S + S_\varepsilon} U_{ij}(P, Q) t_j(Q) dS(Q) \quad (2.37)$$

La partie gauche de l'équation (2.37) est égale à :

$$\lim_{\varepsilon \rightarrow 0} \int_S T_{ij}(P, Q) u_j(Q) dS(Q) + \lim_{\varepsilon \rightarrow 0} \int_{S_\varepsilon} T_{ij}(P, Q) u_j(Q) dS(Q)$$

$$= \int_S T_{ij}(P, Q) u_j(Q) dS(Q) + \lim_{\varepsilon \rightarrow 0} \int_S T_{ij}(P, Q) u_j(Q) dS(Q)$$

Comme précédemment on a :

$$\lim_{\varepsilon \rightarrow 0} \int_{S_\varepsilon} T_{ij}(P, Q) u_j(P) dS(Q) = C_{ij}(P) u_j(P) \quad (2.38)$$

avec $C_{ij}(P) = \lim_{\varepsilon \rightarrow 0} \int_{S_\varepsilon} T_{ij}(P, Q) dS(Q)$

La partie gauche de l'équation (2.37) sera alors égale à :

$$C_{ij}(P) u_j(P) + \int_S T_{ij}(P, Q) u_j(Q) dS(Q) \quad (2.39)$$

La partie droite de l'équation (2.38) est égale à :

$$\lim_{\varepsilon \rightarrow 0} \int_{S + S_\varepsilon} U_{ij}(P, Q) t_j(Q) dS(Q) =$$

$$\lim_{\varepsilon \rightarrow 0} \int_S U_{ij}(P, Q) t_j(Q) dS(Q) + \lim_{\varepsilon \rightarrow 0} \int_{S_\varepsilon} U_{ij}(P, Q) t_j(Q) dS(Q)$$

$$= \int_S U_{ij}(P, Q) t_j(Q) dS(Q) + \lim_{\varepsilon \rightarrow 0} \int_{S_\varepsilon} U_{ij}(P, Q) t_j(Q) dS(Q)$$

comme précédemment la limite :

$$\lim_{\varepsilon \rightarrow 0} \int_{S_\varepsilon} U_{ij}(P, Q) t_j(Q) dS(Q) = 0 \quad (2.40)$$

la partie droite de l'équation (2.37) sera égale à :

$$\int_S U_{ij}(P, Q) t_j(Q) dS(Q) \quad (2.41)$$

finalément on a :

$$C_{ij}(P) u_j(P) + \int_S T_{ij}(P, Q) u_j(Q) dS(Q) = \int_S U_{ij}(P, Q) t_j(Q) dS(Q) \quad (2.42)$$

$$\text{où : } C_{ij}(P) = \lim_{\varepsilon \rightarrow 0} \int_{S_\varepsilon} T_{ij}(P, Q) dS(Q) \quad (2.43)$$

R' étant continue, on a alors :

$$C_{ij}(P) = \delta_{ij}$$

d'où on a :

$$u_i(P) + \int_S T_{ij}(P, Q) u_j(Q) dS(Q) = \int_S U_i(P, Q) t(Q) dS(Q) \quad (2.44)$$

d'une autre façon on a :

$$u_i(P) = - \int_S T_{ij}(P, Q) u_j(Q) dS(Q) + \int_S U_{ij}(P, Q) t_j(Q) dS(Q) \quad (2.45)$$

qui est appelée l'identité de SOMIGLIANA .

Avec :

$$U_{ij}(P, Q) = \frac{(1+\nu)}{8 \pi E (1-\nu) r} \left\{ (3-4\nu) \delta_{ij} + \frac{(P_i - Q_i)(P_j - Q_j)}{r^2} \right\}$$

$$T_{ij}(P,Q) = \frac{1}{8 \pi (1-\nu) r^2} \left\{ (1-2\nu) \left[n_i(Q) \frac{(P_j - Q_j)}{r} - n_j(Q) \frac{(P_i - Q_i)}{r} \right] + n_m(Q) \frac{(P_m - Q_m)}{r} \left[(1-2\nu) \delta_{ij} + \frac{(P_i - Q_i)(P_j - Q_j)}{r^2} \right] \right\}$$

2.8 CALCUL DU TENSEUR DES CONTRAINTES DES POINTS INTERIEURS :

D'après la loi de HOOKE le tenseur des contraintes σ_{ij} est lié au tenseur des déformations par :

$$\sigma_{ij} = \lambda \varepsilon_{kk} \delta_{ij} + 2 \mu \varepsilon_{ij} \quad (2.46)$$

$$\text{Or } \varepsilon_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i}) \quad (2.47)$$

Donc on a :

$$\sigma_{ij} = \lambda \delta_{ij} u_{k,k} + \mu (u_{i,j} + u_{j,i}) \quad (2.48)$$

En injectant la valeur des déplacement $u_i(P)$ dans cette équation on aura :

$$\sigma_{ij}(P) = - \int_S S_{ijk}(P,Q) u_k(Q) dS(Q) + \int_S D_{ijk}(P,Q) t_k(Q) dS(Q) \quad (2.49)$$

Où :

$$S_{ijk}(P,Q) = \lambda \delta_{ij} T_{lk,l}(P,Q) + \mu T_{jk,i}(P,Q) + \mu T_{ik,j}(P,Q) \quad (2.50)$$

$$D_{ijk}(P,Q) = \lambda \delta_{ij} U_{lk,l}(P,Q) + \mu U_{jk,i}(P,Q) + \mu U_{ik,j}(P,Q) \quad (2.51)$$

Soit :

$$\begin{aligned}
S_{ijk}(P, Q) &= \frac{E}{8 \pi (1-\nu) r^3} \\
&\left\{ n_l(Q) \frac{(P_l - Q_l)}{r} \left[3(1-2\nu) \delta_{ij} \frac{(P_k - Q_k)}{r} \right. \right. \\
&\quad \left. \left. + 3\nu \left\{ \delta_{ik} \frac{(P_j - Q_j)}{r} - \delta_{jk} \frac{(P_i - Q_i)}{r} \right\} \right. \right. \\
&\quad \left. \left. - 15 \frac{(P_i - Q_i)(P_j - Q_j)(P_k - Q_k)}{r^3} \right] \right. \\
&+ n_i(Q) \left[3\nu \frac{(P_j - Q_j)(P_k - Q_k)}{r^2} + (1-2\nu) \delta_{jk} \right] \\
&+ n_j(Q) \left[3\nu \frac{(P_i - Q_i)(P_k - Q_k)}{r^2} + (1-2\nu) \delta_{ik} \right] \\
&\left. \left. + n_k(Q) \left[3\nu \frac{(P_i - Q_i)(P_j - Q_j)}{r^2} + (1-2\nu) \delta_{ik} \right] \right\} \quad (2.52)
\end{aligned}$$

Et

$$\begin{aligned}
D_{ijk}(P, Q) &= \frac{1}{8 \pi (1-\nu) r^2} \\
&\left\{ (1-2\nu) \left[\delta_{ij} \frac{(P_k - Q_k)}{r} - \delta_{ik} \frac{(P_j - Q_j)}{r} - \delta_{jk} \frac{(P_i - Q_i)}{r} \right] \right. \\
&\quad \left. - 3 \frac{(P_i - Q_i)(P_j - Q_j)(P_k - Q_k)}{r^3} \right\} \quad (2.53)
\end{aligned}$$

CHAPITRE 3

TRAITEMENT NUMERIQUE DE L' EQUATION INTEGRALE

3.1 INTRODUCTION :

Les solutions analytiques des équations intégrales relatives aux problèmes d'élasticité sont extrêmement rares, et n'existent que pour quelques géométries et conditions aux limites simples .

Pour être en mesure de traiter des cas plus complexes, et correspondant à ceux rencontrés dans la pratique en mécanique, on est amené à utiliser des techniques numériques sur des ordinateurs de grande capacité .

Ces techniques numériques nous permettent d'approcher la solution du problème , avec une erreur qui dépend généralement de:

- Le nombre d'éléments utilisés .
- La représentation de la géométrie .
- la représentation des fonctions .
- La méthode d'intégration .

Dans cette étude on se limite à mettre en évidence l'adaptation de la méthode des équations intégrales aux problèmes tridimensionnels. Pour cela le problème d'évaluation de l'erreur ne sera pas traité .

La surface S de la région R est représentée par N éléments quadratiques .

Chaque élément est défini par quatre noeuds qui délimitent sa surface $(\Delta S)_k$. La numérotation de ces noeuds se fait comme il est indiqué sur la figure (fig 3.1).

La forme du quadrilatère n'est pas nécessairement un rectangle, il peut avoir une forme quelconque .

Les fonctions u_i^k et t_i^k qui représentent respectivement le champ des déplacements et le champ des tractions de l'élément k suivant la direction x_i , sont supposées constantes sur l'élément, et elles sont associées au centroïde P_k de l'élément.

Le centroïde P_k de l'élément $(\Delta S)_k$ est l'intersection des deux droites Δ_1^k et Δ_2^k :

Où Δ_1^k est la droite qui passe par les milieux des segments $[C_1^k, C_2^k]$ et $[C_3^k, C_4^k]$.

Et Δ_2^k est la droite qui passe par les milieux des segments $[C_1^k, C_3^k]$ et $[C_2^k, C_4^k]$.

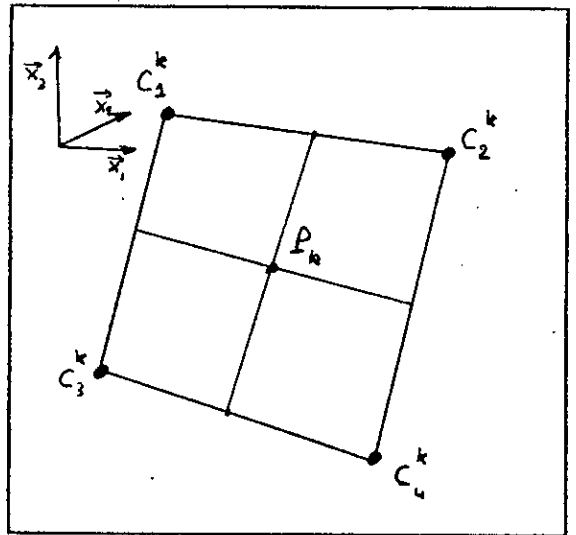


Fig 3.1

Element quadrilatère à quatre noeuds

3.2 DISCRETISATION DE L'EQUATION INTEGRALE :

Dans le chapitre précédent on a abouti à une équation intégrale qui relie directement les paramètres de la surface $(u_i(Q), t_i(Q))$ entre eux (Eq 2.35) :

$$\frac{1}{2} \delta_{ij} u_j(P) + \int_S T_{ij}(P, Q) u_j(Q) dS(Q) = \int_S U_{ij}(P, Q) t_j(Q) dS(Q) \quad (3.1)$$

Où P est un point de la surface là où on applique la force concentrée .

Et Q est un point quelconque de la surface .

Si on discrétise la surface S en N éléments $(\Delta S)_k$ on aura :

$$\frac{1}{2} \delta_{ij} u_j(P_k) + \sum_{l=1}^N \int_{(\Delta S)_l} T_{ij}(P_k, Q) u_j(Q) dS(Q) = \sum_{l=1}^N \int_{(\Delta S)_l} U_{ij}(P_k, Q) t_j(Q) dS(Q) \quad (3.2)$$

Où Q est un point de l'élément $(\Delta S)_k$.

Si on suppose que $u_j(Q)$ et $t_j(Q)$ sont constants sur l'élément $(\Delta S)_l$ et en posant :

$$\begin{cases} u_j(Q) = u_j(P_l) \\ t_j(Q) = t_j(P_l) \end{cases} \quad (Q \in (\Delta S)_l)$$

On aura :

$$\frac{1}{2} \delta_{ij} u_j(P_k) + \sum_{l=1}^N \left\{ \left[\int_{(\Delta S)_l} T_{ij}(P_k, Q) dS(Q) \right] u_j(P_l) \right\} = \sum_{l=1}^N \left\{ \left[\int_{(\Delta S)_l} U_{ij}(P_k, Q) dS(Q) \right] t_j(P_l) \right\} \quad (3.3)$$

Posant :

$$T_{ij}^{kl} = \int_{(\Delta S)_l} T_{ij}(P_k, Q) dS(Q)$$

$$\text{et } U_{ij}^{kl} = \int_{(\Delta S)_l} U_{ij}(P_k, Q) dS(Q)$$

On obtient :

$$\frac{1}{2} \delta_{ij} u_j(P_k) + \sum_{l=1}^N \left\{ T_{ij}^{kl} u_j(P_l) \right\} = \sum_{l=1}^N \left\{ U_{ij}^{kl} t_j(P_l) \right\} \quad (3.4)$$

On a ainsi une seule équation à $6N$ paramètres :

$$(u_j(P_l), t_j(P_l)) \quad l = 1, \dots, N \quad ; \quad j = 1, 2, 3$$

Pour un problème bien posé, la moitié de ces paramètres est donnée comme conditions aux limites et il reste $3N$ inconnues à déterminer.

En faisant varier le point d'application de la force concentrée P_k sur tout les éléments $(\Delta S)_k$ ($k = 1, \dots, N$) et suivant toute les directions x_i ($i = 1, 2, 3$), on obtient un système d'équations de $3N$ équations algébriques à $3N$ inconnues.

$$\left[\begin{array}{l} \frac{1}{2} \delta_{ij} u_j(P_k) + \sum_{l=1}^N \left\{ T_{ij}^{kl} u_j(P_l) \right\} = \sum_{l=1}^N \left\{ U_{ij}^{kl} t_j(P_l) \right\} \\ k = 1, \dots, N \\ i = 1, 2, 3 \end{array} \right. \quad (3.5)$$

NB :

Dans l'équations (3.5), et selon la convention adoptée, il y a une sommation suivant l'indice j.

posons :

$$n = 3(k-1) + i$$

$$n = 3(l-1) + j$$

$$\text{et } A_{mn} = T_{ij}^{kl} + \frac{1}{2} \delta_{ij} \delta_{kl}$$

$$B_{mn} = U_{ij}^{kl}$$

ainsi que $u_n = u_j(P_l)$

$$t_n = t_j(P_l)$$

Le système d'équation (- -) devient :

$$\left[\begin{array}{l} \sum_{n=1}^{3N} A_{mn} u_n = \sum_{n=1}^{3N} B_{mn} t_n \\ n = 1, \dots, 3N \end{array} \right. \quad (3.6)$$

Ou sous une forme matricielle :

$$\left[\begin{array}{c} A \end{array} \right] \left\{ \begin{array}{c} u \end{array} \right\} = \left[\begin{array}{c} B \end{array} \right] \left\{ \begin{array}{c} t \end{array} \right\} \quad (3.7)$$

qui représente un système de $3N$ équations à $6N$ paramètres .

$$\text{avec : } \begin{cases} A_{mn} = T_{ij}^{kl} + \frac{1}{2} \delta_{ij} \delta_{kl} \\ B_{mn} = U_{ij}^{kl} \end{cases}$$

$$\text{Où : } \begin{cases} T_{ij}^{kl} = \int_{(\Delta S)_l} T_{ij}(P_k, Q) dS(Q) \\ U_{ij}^{kl} = \int_{(\Delta S)_l} U_{ij}(P_k, Q) dS(Q) \end{cases}$$

$$\text{et } \begin{cases} k = \left[\frac{m}{3} \right] + 1 \\ i = m - 3(k-1) \\ l = \left[\frac{n}{3} \right] + 1 \\ j = n - 3(l-1) \end{cases} \quad \text{ou } [x] \text{ représente la partie entière de } x$$

3.3 INTEGRATION NUMERIQUE :

C'est l'étape la plus importante dans la méthode des équations intégrales car :

- Pour obtenir une bonne approximation de la solution du problème , il faut déterminer avec précision les coefficients des matrices A et B qui sont les valeurs des intégrales des noyaux $T_{ij}(P,Q)$ et $U_{ij}(P,Q)$ sur les éléments.

- Le nombre de coefficients à déterminer est égal à $(3N \times 3N) \times 2$. soit $(18 N^2)$ coefficients , ce qui prend la majorité du temps calcul lors de l'exécution .

Donc il faut trouver une méthode d'intégration qui est à la fois rapide et précise .

La solution idéale est de les évaluer analytiquement , ce qui est très difficile pour les cas tridimensionnels surtout si on utilise des éléments d'ordres supérieurs

les noyaux $T_{ij}(P,Q)$ et $U_{ij}(P,Q)$, ont été intégrés numériquement, moyen qui peut être facilement appliqué aux éléments d'ordres supérieur.

3.3.1 CHOIX DE LA METHODE NUMERIQUE D'INTEGRATION :

Les méthodes d'intégrations sur une surface, les plus précises sont :

- La méthode de quadrature de GAUSS .
- La méthode de NEWTON-COTES .

La méthode de quadrature de GAUSS détermine exactement la valeur de l'intégrale d'un polynome de degré $(2N-1)$ en calculant seulement ces valeurs en N points, alors que la méthode de NEWTON-COTES nécessite $(2N+1)$ points pour faire le même travail.

Pour cette raison le schémas d'intégration numérique qui sera utilisé est celui de la méthode de GAUSS.

3.3.2 METHODE DE QUADRATURE DE GAUSS :

Soit F une fonction continue, à deux variables, alors d'après la formule de quadrature de gauss on a :

$$\int_{-1}^1 \int_{-1}^1 F(r,s) dr ds \cong \sum_{i=1}^{NPG} \sum_{j=1}^{NPG} \alpha_i \alpha_j F(r_i, s_j) \quad (3.8)$$

Où NPG : est le nombre de point de gauss retenu

α_i, α_j : coefficients tabulés ou poids de gauss

r_i, s_i : coordonnées des points de gauss.

Pour évaluer : $I = \int_{(\Delta S)_l} T(P_{ij}, Q_k) dS(Q)$ on procède comme suit :

1) Si $k \neq l$: Le centroïde de l'élément $(\Delta S)_l$ est différent du point d'application de la force P_k .

L'élément $(\Delta S)_l$ est défini par quatre points $C_1^l, C_2^l, C_3^l, C_4^l$, dont les coordonnées sont données dans le repère global \vec{x}_i , $i=1,2,3$.

En effectuant un changement de repère, on aura les coordonnées de ces quatre points dans le plan de l'élément.

Donc on peut écrire :

$$I = \int_{(\Delta S)_l} T_{ij}(P_k, Q) dx dy$$

En utilisant l'élément isoparamétrique à quatre noeuds on aura :

$$x = \frac{1}{4} [(1+r)(1+s)x_1 + (1+r)(1+s)x_2 + (1+r)(1+s)x_3 + (1+r)(1+s)x_4]$$

$$y = \frac{1}{4} [(1+r)(1+s)y_1 + (1+r)(1+s)y_2 + (1+r)(1+s)y_3 + (1+r)(1+s)y_4]$$

$$\text{et } I = \int_{-1}^1 \int_{-1}^1 T_{ij}(P_k, Q) \det(J) dr ds$$

$$\text{Où } \det(J) = \frac{\partial x}{\partial r} \frac{\partial y}{\partial s} - \frac{\partial y}{\partial r} \frac{\partial x}{\partial s}$$

$$\text{avec : } \frac{\partial x}{\partial r} = \frac{1}{4} [(1+s)x_1 - (1+s)x_2 - (1-s)x_3 + (1-s)x_4]$$

$$\frac{\partial x}{\partial s} = \frac{1}{4} [(1+r)x_1 + (1-r)x_2 - (1-r)x_3 - (1+r)x_4]$$

$$\frac{\partial y}{\partial r} = \frac{1}{4} [(1+s)y_1 - (1+s)y_2 - (1-s)y_3 + (1-s)y_4]$$

$$\frac{\partial y}{\partial s} = \frac{1}{4} [(1+r)y_1 + (1-r)y_2 - (1-r)y_3 - (1+r)y_4]$$

alors on aura :

$$I = \sum_{m=1}^{NPG} \sum_{n=1}^{NPG} \alpha_m \alpha_n T_{ij}(P_{kmn}, Q_{mn}) \det(J(r_m, s_n))$$

Les coordonnées des points P_{kmn} et Q_{mn} dans le repère global seront déterminées à partir de r_m et s_n :

(r_m, s_n) sont les coordonnées des points dans le repère lié à l'élément isoparamétrique. A partir de (r_m, s_n) on détermine les coordonnées des points dans le repère local (x, y) , et avec une transformation inverse on détermine les coordonnées des points dans le repère global, $(\vec{x}_1, \vec{x}_2, \vec{x}_3)$.

La même procédure est faite pour déterminer : $\int_{(\Delta S)_k} U_{ij}(P_k, Q) dS(Q)$

2) Si $k = 1$: Le point d'application de la force concentrée est le centroïde de l'élément.

En ce point P_k $T_{ij}(P_k, Q)$ ainsi que $U_{ij}(P_k, Q)$ tendent vers l'infini. Donc ces deux fonctions ne sont pas continues sur $(\Delta S)_k$ et par conséquent ne peuvent pas être approximer par des polynôme: l'utilisation directe de la méthode de GAUSS n'est plus possible.

Pour contourner ce problème on isole le centroïde P_k par un carré de coté égale a 2ε (ε très petit) et divise l'élément $(\Delta S)_k$ en quatre sous-éléments $(\Delta S)_k^p$ ($p = 1, 2, 3, 4$) .fig(3.2)

et on aura par exemple :

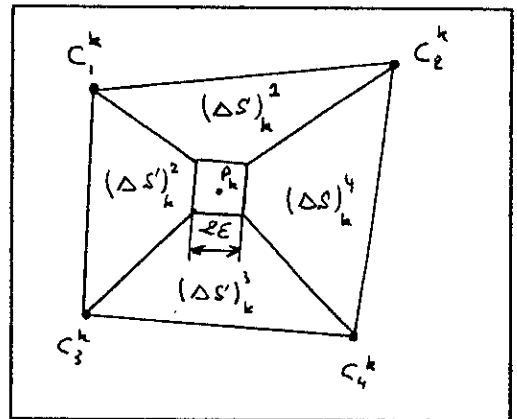


fig 3.2
Forme des quatres sous-éléments de l'élément (ΔS)

$$\int_{(\Delta S)_k} T_{ij}(P_k, Q) dS(Q) = \sum_{p=1}^4 \int_{(\Delta S)_k^p} T_{ij}(P_k, Q) dS(Q) \quad (3.9)$$

$T_{ij}(P_k, Q)$ étant continu sur $(\Delta S)_k^p$, on peut donc appliquer la méthode de quadrature de GAUSS.

Cette technique donne une très bonne précision si ε est très petit devant les dimensions de l'élément. Cela a été vérifié pour le calcul de l'intégrale du noyau $T_{11}(P_1, Q)$, pour différents nombres de points de GAUSS. voir fig(3.3).

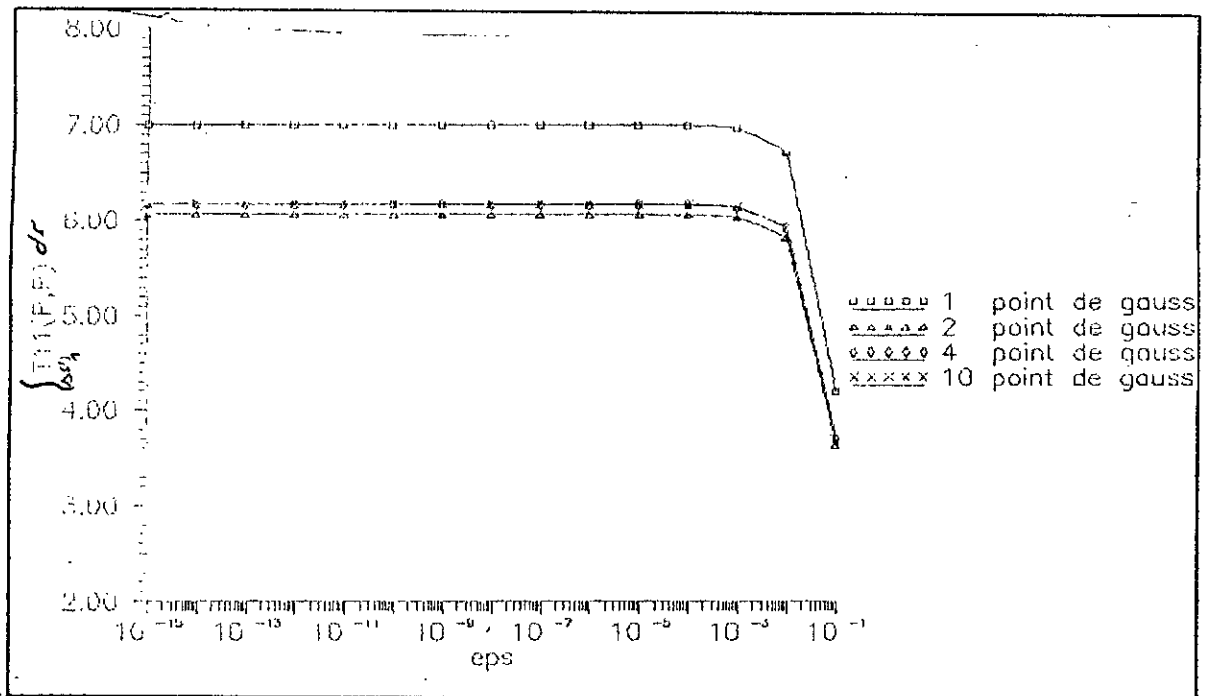


fig 3.3 INFLUENCE DE ϵ SUR LA VALEUR DE L'INTEGRALE DE $T_{ij}(P,P)$

3.3.3 CHOIX DU NOMBRE DE POINTS DE GAUSS :

Le choix du nombre de point de gauss est très important car :

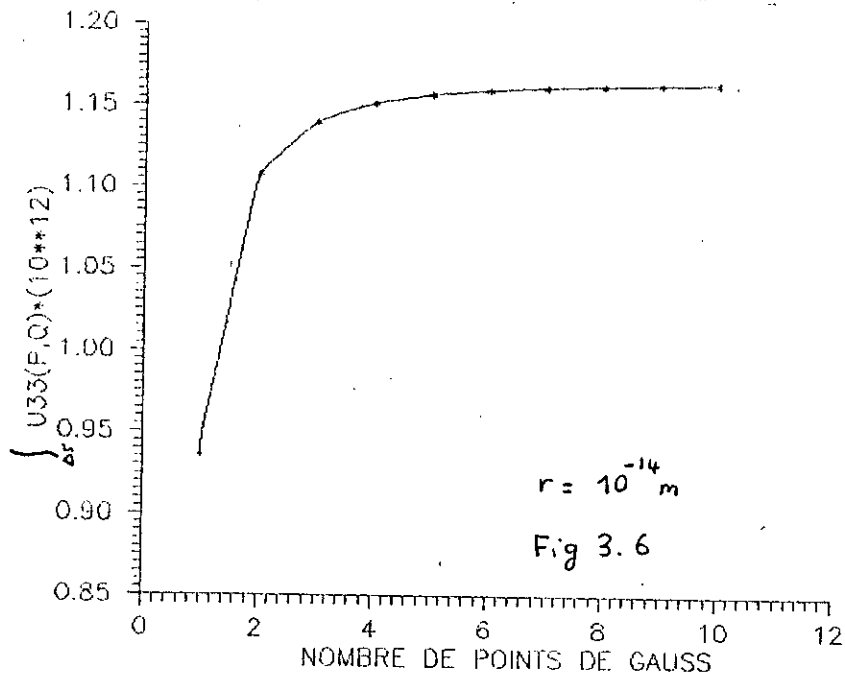
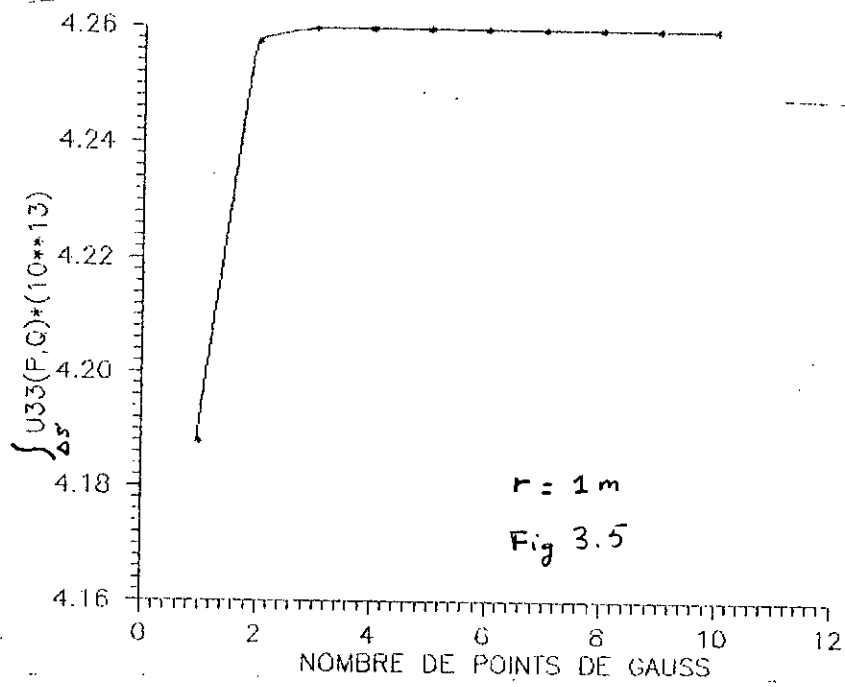
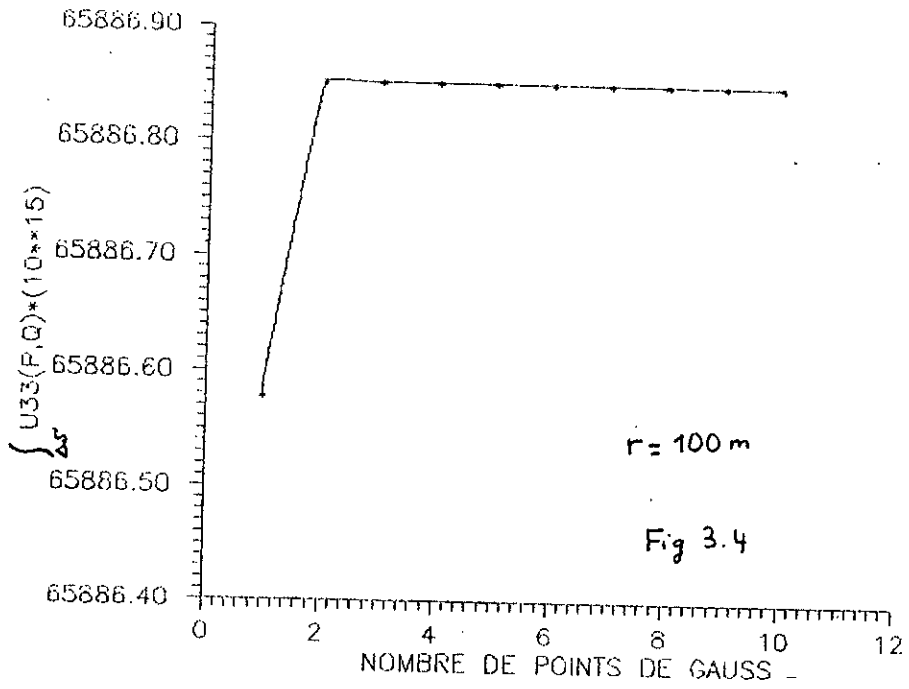
1- Le coût de l'analyse augmente considérablement avec le nombre de points de GAUSS : Soit t le temps nécessaire pour calculer l'intégrale avec un seule point de GAUSS, alors si on utilise N point le temp calcul sera: $t_N = t N^2$

donc on a intérêt à choisir un nombre de points le minimale possible .

2- En utilisant des ordres d'intégrations différents , les resultats peuvent être très affectés , donc il faut utiliser l'ordre le plus élevé pour avoir une bonne approximation .

En générale , le nombre de points de GAUSS dépend de la fonction à évaluer : Si cette fonction varie rapidement sur le domaine d'intégration il faut prendre un nombre de points plus élevé ,dans le cas contraire où la fontions est plus ou moin constante sur le domaine d'intégration il faut utiliser un nombre réduit de points de GAUSS .

INFLUENCE DU NOMBRE DE POINTS DE GAUSS



Ceci a été vérifié pour l'intégrale du noyau $U_{93}(P,Q)$:

La figure fig(3.4) représente la valeur de l'intégrale en fonction du nombre de points de GAUSS lorsque la singularité est très loin du domaine d'intégration ($r = 100 \text{ m}$) ; deux points suffisent pour évaluer l'intégrale .

Dans la figure fig(3.5) le domaine d'intégration est à une distance moyenne de la singularité ($r = 1 \text{ m}$) ; quatre points sont nécessaires pour obtenir une bonne approximation .

Dans la figure fig(3.6) le domaine d'intégration est très proche de la singularité ($r = 10^{-14} \text{ m}$) ; plus de huit points de GAUSS pour avoir une bonne approximation .

Dans le programme M.I.T.D. , les intégrales sont évaluées à dix chiffres après la virgule , Pour lesquelles faut 6 points de gauss si $k \neq 1$ et 8 points de gauss si $k = 1$.

3.4 CONSTITUTION DU SYSTEME D'EQUATION :

3.4.1 INTRODUCTION DES CONDITIONS AUX LIMITES :

Dans le paragraphe précédent on est arrivé à un système d'équations algébrique de $3N$ équations à $6N$ paramètres (Eq 3.7) :

$$\begin{bmatrix} A \end{bmatrix} \begin{Bmatrix} u \end{Bmatrix} = \begin{bmatrix} B \end{bmatrix} \begin{Bmatrix} t \end{Bmatrix} \quad (3.10)$$

$3N$ de ces paramètres sont données comme conditions aux limites, on a trois types de problèmes :

1) PROBLEME DE DIRICHLET :

Le champ des déplacements u_n ($n = 1, \dots, 3N$) est connu , alors le système d'équations à résoudre (3.10) devient :

$$\begin{bmatrix} B \end{bmatrix} \begin{Bmatrix} t \end{Bmatrix} = \begin{Bmatrix} f \end{Bmatrix} \quad \text{où} \quad \begin{Bmatrix} f \end{Bmatrix} = \begin{bmatrix} A \end{bmatrix} \begin{Bmatrix} u \end{Bmatrix} \quad (3.11)$$

La résolution de ce système nous permet d'avoir le champ des tensions t_n ($n = 1, \dots, 3N$) sur la surface S.

2) PROBLEME DE NEUMAN :

Le champ des contraintes t_n ($n=1, \dots, 3N$) est connu ,alors on aura à résoudre :

$$\begin{bmatrix} A \end{bmatrix} \begin{Bmatrix} u \end{Bmatrix} = \begin{Bmatrix} f \end{Bmatrix} \quad \text{où} \quad \begin{Bmatrix} f \end{Bmatrix} = \begin{bmatrix} B \end{bmatrix} \begin{Bmatrix} t \end{Bmatrix} \quad (3.12)$$

La résolution de ce système nous permet d'avoir le champ des déplacements u_n ($n = 1, \dots, 3N$) sur la surface S.

3) PROBLEME MIXTE :

Deux types de conditions limites sont connues sur la surface S. Les déplacements sont connus sur une partie de S et des contraintes sont connus sur l'autre partie .

En mettant les inconnues d'un seul coté on obtient un système d'équations de la forme :

$$\begin{bmatrix} K \end{bmatrix} \begin{Bmatrix} x \end{Bmatrix} = \begin{Bmatrix} f \end{Bmatrix} \quad (3.13)$$

La résolution de ce système nous donne les paramètres inconnus de la surface S.

3.5 RESOLUTION DU SYSTEME D'EQUATION :

3.5.1 CONDITIONNEMENT DE MATRICE :

Les coefficients des matrices A et B ne sont pas du même ordre de grandeur :

$$A_{mn} \sim 10^{-2}$$

$$B_{mn} \sim 10^{-11}$$

Pour obtenir un système d'équation bien conditionné ,il faut que les coefficients des deux matrices aient le même ordre de grandeur .

Pour contourner ce problème on multiplie les coefficients de B par le module de cisaillement $G \times 2$. à la fin de la résolution on doit multiplier les valeurs des tensions trouvées par $(2 \times G)$ pour avoir la solution du problème .

$$\begin{bmatrix} A \end{bmatrix} \begin{Bmatrix} u \end{Bmatrix} = \begin{bmatrix} B' \end{bmatrix} \begin{Bmatrix} t' \end{Bmatrix} \quad (3.14)$$

$$\text{où } B'_{mn} = B_{mn} \times 2G \quad \text{et } t' = \frac{t}{2G}$$

3.5.2 CHOIX DE LA METHODE DE RESOLUTION :

La matrice [K] du système à résoudre n'est pas symétrique et elle est très peuplée . De plus les éléments de la diagonale (l'effet des éléments sur eux mêmes) ne sont pas nuls .

Pour résoudre ce système on a choisi la méthode de résolution LU qui est une méthode directe, pour les raisons suivantes :

- Les méthodes itératives sont écartées car la taille de la matrice [K] ne dépasse pas 500 , au delà de cette limite les méthodes itératives sont les plus efficaces .
- Le temp calcul de la méthode LU est comparable à celui de GAUSS $\sim t \times N^3$.
- La méthode LU est plus précise que la méthode de gauss pour une taille supérieure à 100.

3.6 DETERMINATION DES DEPLACEMENTS DES POINTS INTERIEURS :

Soit P un point où l'on veut connaître le vecteur déplacement. Alors d'après l'équations de SOMIGLIANA (Eq 2.45) :

$$u_i(P) = - \int_S T_{ij}(P,Q) u_j(Q) dS(Q) + \int_S U_{ij}(P,Q) t_j(Q) dS(Q)$$

$$= - \sum_{l=1}^N \int_{(\Delta S)_l} T_{ij}(P, Q) u_j(Q) dS(Q) + \sum_{l=1}^N \int_{(\Delta S)_l} U_{ij}(P, Q) t_j(Q) dS(Q) \quad (3.15)$$

Si on considère que $u_j(Q)$ et $t_j(Q)$ sont constants sur l'élément $(\Delta S)_l$ alors on posant $u_j(Q) = u_j(Q_l)$ et $t_j(Q) = t_j(Q_l)$ sur cet élément on a :

$$u_i(P) = - \sum_{l=1}^N \left\{ \left[\int_{(\Delta S)_l} T_{ij}(P, Q) dS(Q) \right] u_j(Q_l) \right\} + \sum_{l=1}^N \left\{ \left[\int_{(\Delta S)_l} U_{ij}(P, Q) dS(Q) \right] t_j(Q_l) \right\} \quad (3.16)$$

En évaluant les valeurs des intégrales :

$$\int_{(\Delta S)_l} T_{ij}(P, Q) dS(Q) \quad \text{et} \quad \int_{(\Delta S)_l} U_{ij}(P, Q) dS(Q)$$

et en faisant une simple sommation on obtient les composantes du vecteur déplacement en un point P qu'on précise préalablement .

3.7 DETERMINATION DES TENSEURS DES CONTRAINTES DES POINTS INTERIEURS

Soit P un point où l'on veut connaître le tenseur des contraintes, alors d'après l'équations (2.49) on a :

$$\sigma_{ij}(P) = - \int_S S_{ijk}(P, Q) u_k(Q) dS(Q) + \int_S D_{ijk}(P, Q) t_k(Q) dS(Q) \\ = - \sum_{l=1}^N \int_{(\Delta S)_l} S_{ijk}(P, Q) u_k(Q) dS(Q) + \sum_{l=1}^N \int_{(\Delta S)_l} D_{ijk}(P, Q) t_k(Q) dS(Q) \quad (3.17)$$

Si on considère que $u_j(Q)$ et $t_j(Q)$ sont constants sur l'élément $(\Delta S)_l$ alors en posant $u_j(Q) = u_j(Q_l)$ et $t_j(Q) = t_j(Q_l)$ sur cet élément on a :

$$\sigma_{ij}(P) = - \sum_{l=1}^N \left\{ \left[\int_{(\Delta S)_l} S_{ijk}(P, Q) dS(Q) \right] u_k(Q_l) \right\} + \sum_{l=1}^N \left\{ \left[\int_{(\Delta S)_l} D_{ijk}(P, Q) dS(Q) \right] t_k(Q_l) \right\} \quad (3.18)$$

En évaluant les valeurs des intégrales :

$$\int_{(\Delta S)_l} S_{ijk}(P, Q) dS(Q) \quad \text{et} \quad \int_{(\Delta S)_l} D_{ijk}(P, Q) dS(Q)$$

et en faisant une simple sommation on obtient les composantes du tenseur des contraintes en un point P qu'on précise préalablement.

CHAPITRE 4

ORGANISATION ET CARACTERISTIQUES GENERALES DU PROGRAMME

4.1- CARACTERISTIQUES DU PROGRAMME :

- Le programme M.E.I.T.D. (Méthode des Equations Intégrales en Trois Dimensions) a pour but de déterminer le champ des déplacements et le champ des contraintes dans un corps dont le volume est fini, ou un corps infini mais de surface limitée (exemple: un trou sphérique dans un corps infini).

- La forme géométrique du corps est définie par des éléments quadratiques de formes quelconques (pas nécessairement des rectangles).

- Les noeuds sont données en coordonnées cartésiennes seulement, pour cela, il y a une nécessité à développer le programme pour qu'il s'adapte aux types de coordonnées cylindriques et sphériques, cela est possible en introduisant une subroutine qui fait la transformation des coordonnées cylindriques ou sphériques en coordonnées cartésiennes.

- Les conditions aux limites doivent être constantes sur chaque élément, et données dans le repère global : Les conditions du type déplacements constants sur une direction autre que les trois directions du repère globale n'est pas acceptée par ce programme. voir (fig 4.1)

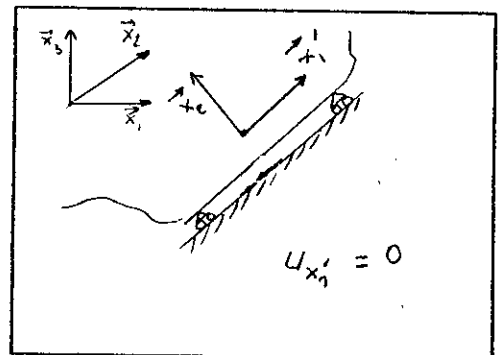


fig 4.1
Déplacement nul dans une direction quelconque

- La taille maximale des matrices [A] et [B] est de 162, donc il permet l'utilisation de 54 éléments seulement, pour cette taille des matrices le fichier exécutable dépasse 500 KO.

- 70% du temps calcul est consacré à l'évaluation des coefficients des matrices [A] et [B] ; une transformation du système à résoudre a été utilisée pour réduire ce temps calcul :

Le système à résoudre étant :

$$\begin{bmatrix} A \end{bmatrix} \begin{Bmatrix} u \end{Bmatrix} = \begin{bmatrix} B \end{bmatrix} \begin{Bmatrix} t \end{Bmatrix} \quad (4.1)$$

En multipliant les deux membres par $8 \pi (1-\nu)$ on aura :

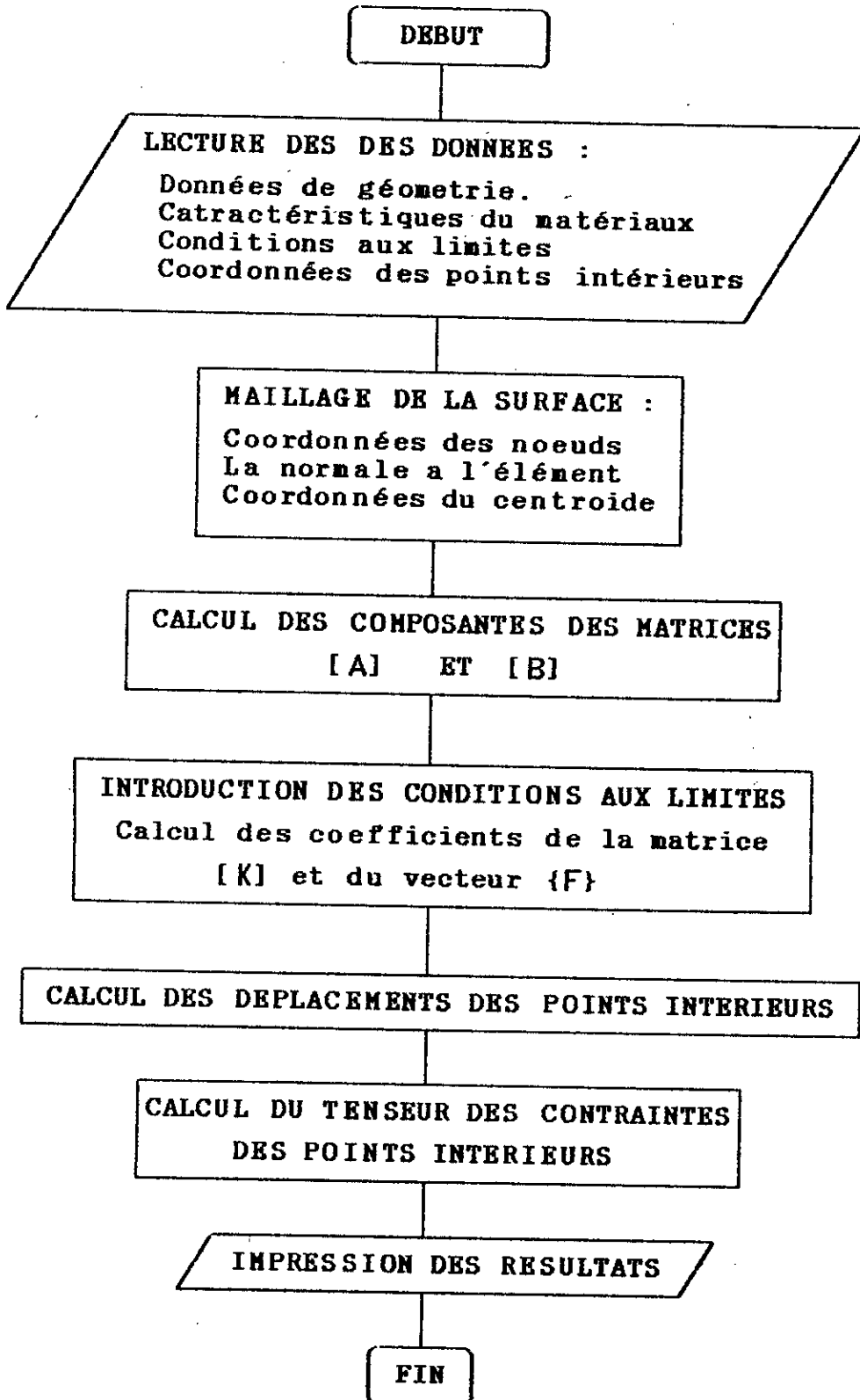
$$8 \pi (1-\nu) \begin{bmatrix} A \end{bmatrix} \begin{Bmatrix} u \end{Bmatrix} = 8 \pi (1-\nu) (2G) \begin{bmatrix} B \end{bmatrix} \begin{Bmatrix} \frac{t}{2G} \end{Bmatrix} \quad (4.2)$$

Ce qui nous permet de diminuer le nombre d'opération par :
 $(5+3) (NPG)^2 N^2$ opérations où NPG : nombre de point de gauss
 N : taille des matrices .

soit 10% du temps calcul

Une transformation similaire est utilisée pour l'évaluation des déplacements et des contraintes des points intérieurs .

4.2- ORGANIGRAMME :



4.3- SUBROUTINES DU PROGRAMME :

Le programme M.E.I.T.D est écrit en FORTRAN 77 , et il contient 23 sous-routines :

1- SUBROUTINE LECTURE :

Elle fait la lecture des données :

a) Lecture des données de géométrie du corps à étudier :

- nn : nombre de noeuds qui définissent la géométrie du corps.
- nplan : nombre de plan qui définissent la géométrie du corps.
- Lecture des coordonnées de chaque noeud par ordre de 1 à nn
- pour chaque plan de 1 à nplan :

Lecture des numéros des quatre noeuds qui le définissent en respectant l'ordre entre ces quatre noeuds (voir fig 4.2)

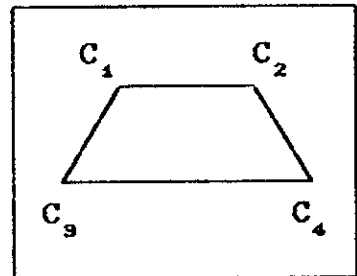


fig 4.2
Ordre des quatre noeuds

- Lecture du nombre d'élément à partager le plan dans la direction C₁C₂ .
- Lecture du nombre d'élément à partager le plan dans la direction C₁C₃ .

b) Lecture des caractéristiques du matériaux :

- E : module de young .
- ν : coefficient de poisson .

c) Lecture des conditions aux limites :

- ndepl : nombre de conditions sur les déplacements .
- Pour chaque condition de déplacement de 1 à ndepl :

- sa valeur algébrique .
 - numero de l'élément où elle est spécifiée
 - sa direction : 1,2 ou 3.
-
- nchar : nombre de conditions de chargements non nuls .
 - pour chaque condition de chargement de 1 à nchar :
 - sa valeur algébrique .
 - numero de l'élément où elle est spécifiée
 - sa direction : 1,2 ou 3.

NB : il n'est pas nécessaire de spécifier un chargement nul .

d) Lecture des coordonnées des points intérieurs :

- npi : nombre de point interieurs
- pour chaque points de 1 à npi :
 - Lecture des coordonnées dans le repère globale en coordonnées cartésiennes.

2 SUBROUTINE MAILLEUR :

Elle permet de faire le maillage des plans qui définissent le corps à étudier : pour chaque élément :

- elle lui donne un numero .
- elle détermine les coordonnées des quatre points qui le définissent.
- elle détermine les coordonnées de son centroïde .
- les coordonnées du vecteur normal à sa surface de telle façon que $\overrightarrow{C_1C_3}$, $\overrightarrow{C_1C_2}$ et \overrightarrow{n} forment une base directe .

La taille des éléments est définie par l'utilisateur .

3 SUBROUTINE MAIPLAN :

C'est une subroutine qui est appelée par le mailleur, elle permet de faire le maillage d'un plan défini dans l'espace.

4 SUBROUTINE ActB :

Elle permet de déterminer les coefficients A_{mn} et B_{mn} des matrices [A] et [B] .

en premier lieu on détermine les indices k,l et i,j :

A_{mn} représente l'intégrale sur l'élément l de l'effet, dans une direction j , d'une force concentrée égale à l'unité, appliqué sur l'élément k dans la direction i .

si $k \neq l$: on fait une simple intégration sur l'élément l avec 6 points de gauss .

si $k = l$: il y a une singularité, on isole le centroïde de l'élément en le divisant en quatre sous-éléments , et on fait une intégration sur chaque sous-éléments , avec huit points de gauss .

5) SUBROUTINE KetF :

En introduisant les conditions aux limites , elle transforme le système d'équations :

$$\begin{bmatrix} A \end{bmatrix} \begin{Bmatrix} u \end{Bmatrix} = \begin{bmatrix} B \end{bmatrix} \begin{Bmatrix} t \end{Bmatrix} \quad (4.3)$$

En un système :

$$\begin{bmatrix} K \end{bmatrix} \begin{Bmatrix} x \end{Bmatrix} = \begin{Bmatrix} f \end{Bmatrix} \quad (4.4)$$

où : $\begin{Bmatrix} x \end{Bmatrix}$ représente les inconnues

6) SUBROUTINE RESLU :

Elle permet la résolution d'un système d'équation $[K]\{X\}=\{F\}$ avec la méthode L U .

La matrice L occupe la partie inférieure de [K] et la matrice U la partie supérieure de [K] .

Le vecteur solution $\{X\}$ occupe le vecteur $\{F\}$.
Il faut que les éléments de la diagonale ne soient pas nuls (ce qui est toujours le cas pour la matrice $[K]$).

7) SUBROUTINE REARG :

Le vecteur solution $\{X\}$ contient des solutions de type déplacement et de type tension, cette subroutine range les déplacements dans un seul vecteur, et les tensions dans l'autres.

8) SUBROUTINE DPINT :

Elle permet le calcul du vecteur déplacement des points intérieurs.

9) SUBROUTINE CPINT :

Elle permet le calcul du tenseur des contraintes des points intérieurs.

10) SUBROUTINE AFFICH :

Elle met les résultats dans un fichier :

- Pour chaque élément :

- Les coordonnées des 4 points qui le définissent.
- La normale à cet élément.
- Les coordonnées du centroïde de l'élément.

- Le champ des déplacements :

pour chaque élément de la surface, les composantes du vecteur déplacement.

- Le champ des tensions :

pour chaque éléments de la surface, les composantes du vecteur tension

- Les vecteurs déplacements des points intérieurs :
pour chaque point :
 - Ces coordonnées .
 - Les composantes du vecteur déplacement .
- Les tenseurs des contraintes des points intérieurs :
pour chaque point :
 - Ces coordonnées .
 - Le tenseur des contraintes

11) SUBROUTINE INTEGRA :

C'est le cœur de ce programme , elle permet de calculer l'intégrale d'une fonction sur un plan défini dans l'espace , avec la méthode de quadrature de GAUSS .

Le nombre de point de GAUSS est choisi par le programmeur , il varie de 1 à 10 .

Pour les points de 1 à 4 , les coefficients sont donnés à 16 chiffres après la virgule , et pour les points de 5 à 10 ils sont donnés à 10 chiffres après la virgule seulement .

12) SUBROUTINE NOYAUT :

C'est le noyau $T_{ij}(P,Q)$ multiplié par $8 \pi (1-\nu)$.

13) SUBROUTINE NOYAUU :

C'est le noyau $U_{ij}(P,Q)$ multiplié par $8 \pi (1-\nu)/(1+\nu)$

14) SUBROUTINE NOYAUD

C'est le noyau $D_{ijk}(P,Q)$ multiplié par $8 \pi (1-\nu)$

15) SUBROUTINE NOYAUS :

C'est le noyau $S_{ijk}(P,Q)$ multiplié par $8 \pi (1-\nu) / E$

16) SUBROUTINE INTERSEC :

Elle détermine les coordonnées du point d'intersection , dans l'espace , de deux droites définies chacune par deux points .

17) SUBROUTINE MATPASS :

Elle détermine la matrice de passage , du repère global au repère local lié à l'élément . Voir (fig 4.3)

Le repère local est défini par :

- son origine : le point C_1 .
- \vec{e}_1 : le premier vecteur de la

base
$$\frac{\overrightarrow{C_1 C_3}}{\left\| \overrightarrow{C_1 C_3} \right\|}$$

- \vec{e}_3 : le troisième vecteur de la base , c'est le vecteur unité normale à l'élément .

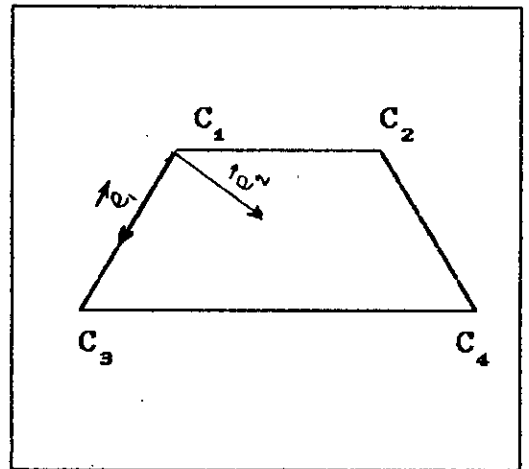


fig 4.3

Repère local lié à l'élément

- \vec{e}_2 : le second vecteur de la base : $\vec{e}_2 = \vec{e}_3 \times \vec{e}_1$

18) SUBROUTINE PRODMV :

Elle fait le produit d'une matrice avec un vecteur .

19) SUBROUTINE PRODVEC :

Elle détermine le vecteur défini par le produit vectoriel de deux vecteurs divisé par sa norme :

$$\vec{w}_3 = \frac{\overrightarrow{w}_1 \times \overrightarrow{w}_2}{\left\| \overrightarrow{w}_1 \times \overrightarrow{w}_2 \right\|}$$

20) FUNCTION CHRON :

C'est la fonction de chronometer , elle vaut 1 pour $i=j$ et 0 pour les cas contraires .

21) SUBROUTINE DIS :

Elle calcule la distance entre deux points P,Q .

22) SUBROUTINE INVERSE :

Elle permet le calcul de l'inverse d'une matrice 3×3 .

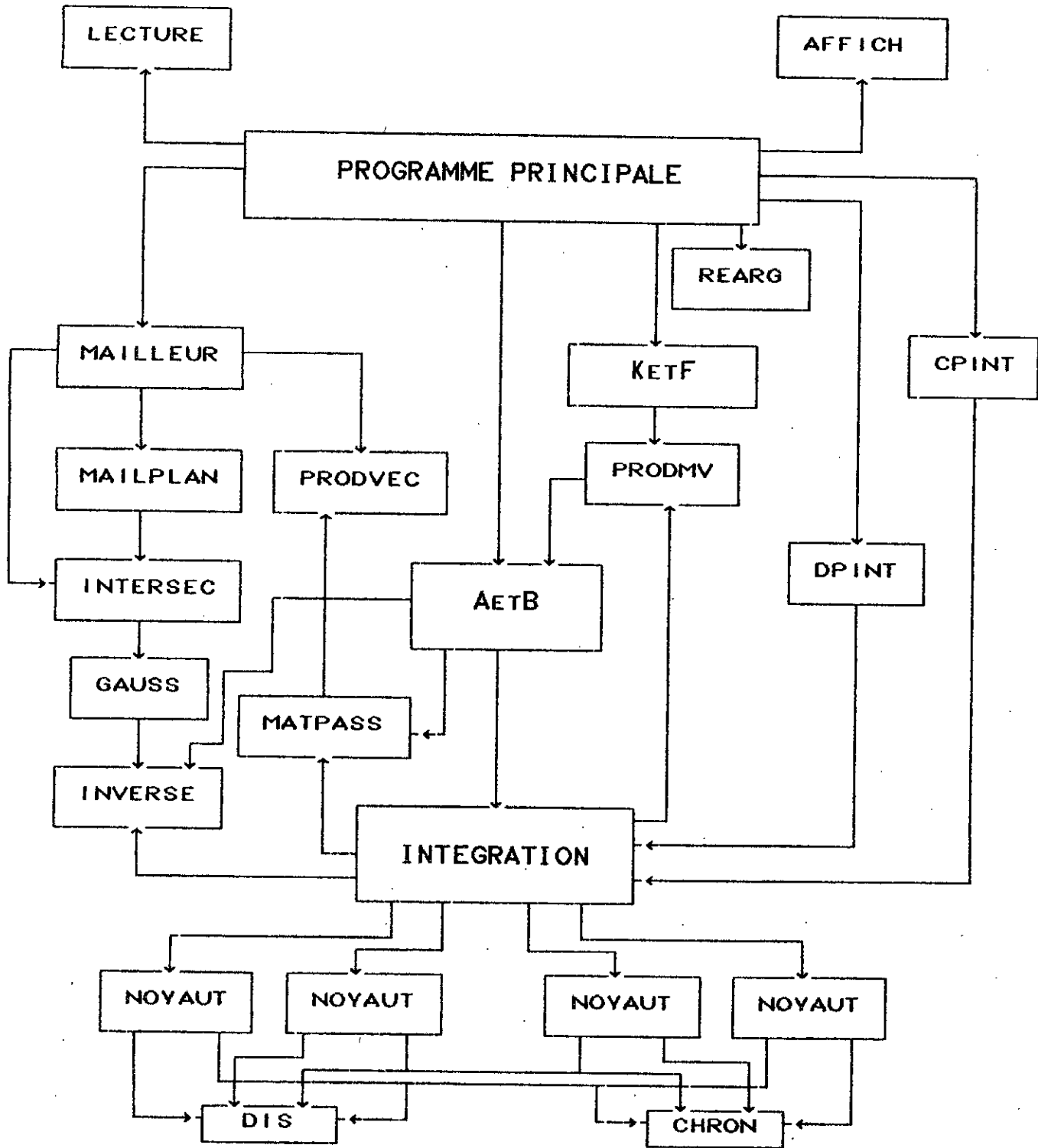
23) SUBROUTINE GAUSS :

Elle permet la résolution des systèmes d'équations d'ordre N , avec la méthode de GAUSS avec pivotation totale .

Elle est utilisée pour calculer l'inverse de la matrice de passage , ainsi que pour déterminer les points d'intersections entre les droites .

Elle n'est pas utilisée pour la résolution du système $[K] \{X\} = \{F\}$ car elle prend plus de temps que la méthode L U .Et elle est moins précise pour les systèmes de grande tailles ($N > 100$).

4.4 SCHEMA REPRESENTANT LES LIAISONS ENTRE LES SUBROUTINES :



CHAPITRE 5

EXEMPLES

Dans ce présent chapitre on va étudier quelque exemples simples . Le but est : - de vérifier la validité du programme M.E.I.T.D.

- de mettre en évidence l'adaptation de la méthode des équations intégrales aux problèmes tridimensionnels .

Vu le matériel informatique disponible, utilisé pour l'exécution du programme (PC 386 , dont le système d'exploitation est le DOS 5.1 qui ne permet pas de gérer un fichier exécutable de taille supérieur à 640KO), on ne peut mailler la surface du corps par plus de 54 éléments .

Le corps utilisé pour ces exemples est un parallélépipède de dimensions $10 \times 10 \times 10 \text{ mm}^3$ (fig 5.1) , son module de YOUNG est E et son coefficient de POISSON est ν .

On s'intéresse aux déplacements et aux contraintes des points P de la droite ($x = 5 \text{ mm}$, $y = 5 \text{ mm}$ et $0 < z < 10 \text{ mm}$) voir fig(5.2)

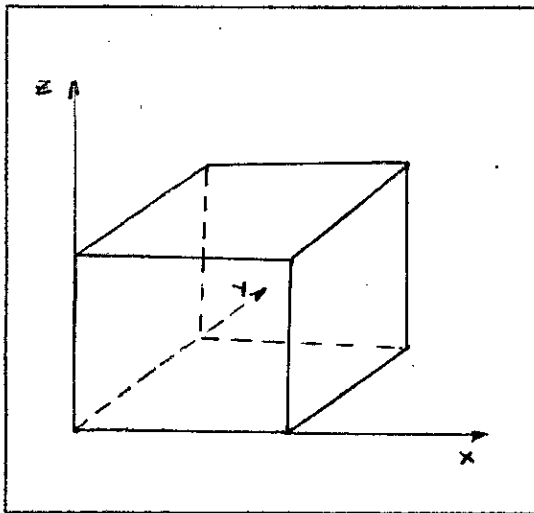


fig 5.1 le cube à étudier dans le repère globale

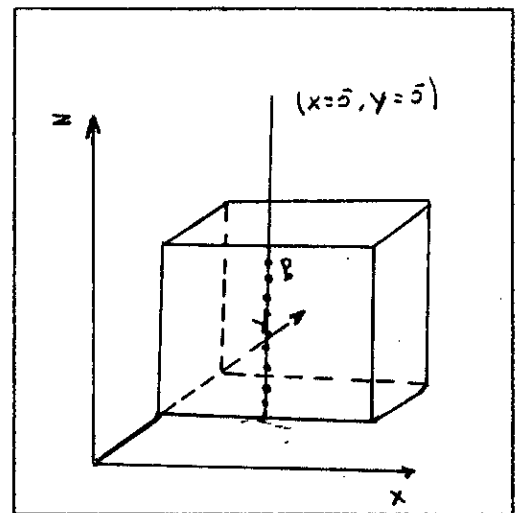


fig 5.2 représentation de la droite $x = 5$, $y = 5$

La figure 5.3 représente le développement sur un plan de la surface S du cube maillé avec 24 éléments et la figure 5.4 représente son développement avec 54 éléments .

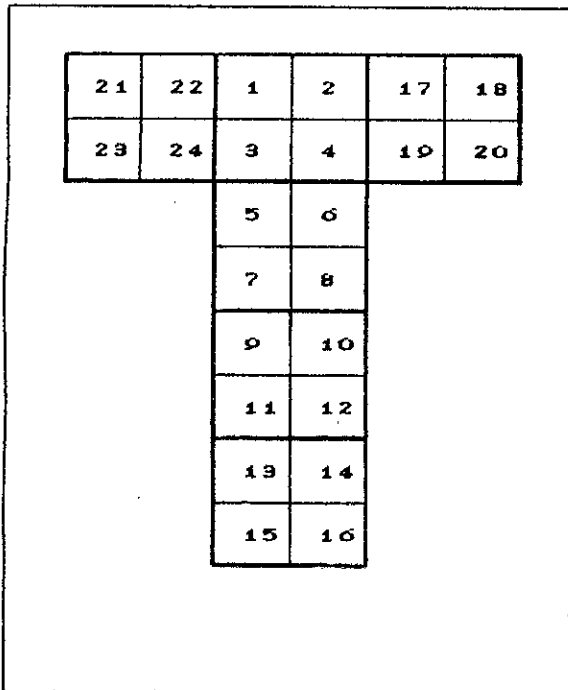


fig 5.3 développement de la surface du cube sur un plan maillage avec 24 éléments

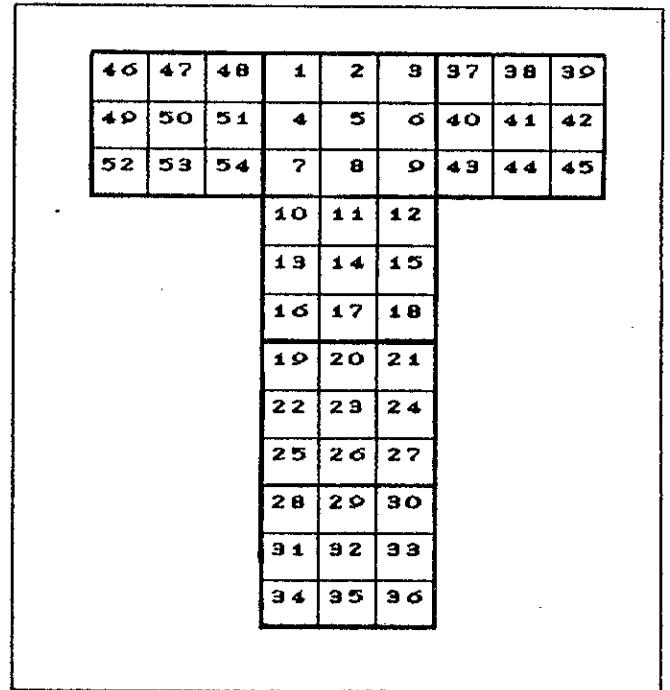


fig 5.4 développement de la surface du cube sur un plan Maillage avec 54 éléments

5.1 DEPLACEMENT DU CORPS RIGIDE .

On soumet le cube à un déplacement du corps rigide :

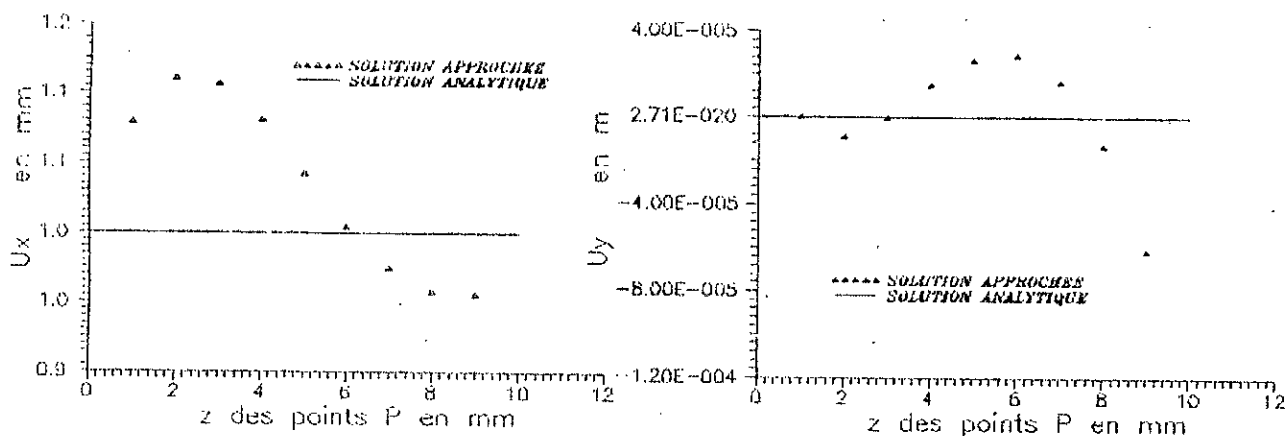
EXEMPLE 1 :

On pose comme conditions aux limites :

$U_x = 1 \text{ mm}$ $U_y = U_z = 0$ pour tout les éléments de la surface .

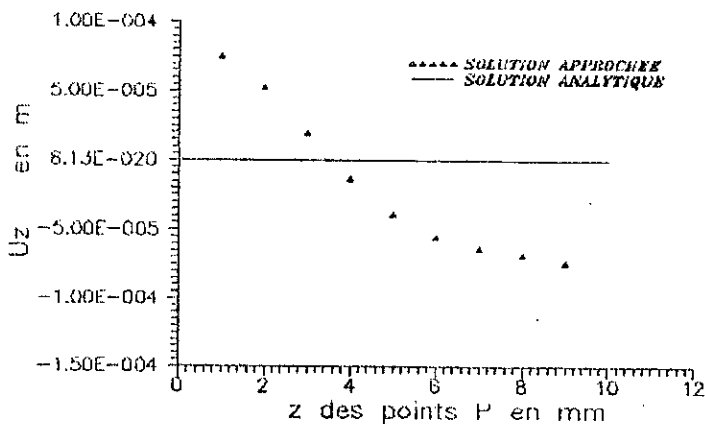
La figure 5.5 représente les composantes du vecteur déplacement des points P de la droite ($x = 5$, $y = 5$) en fonction de z , pour une discrétisation de la surface avec 24 éléments .

La figure 5.6 représente aussi les composantes du vecteur déplacement des points P pour une discrétisation de la surface avec 54 éléments .



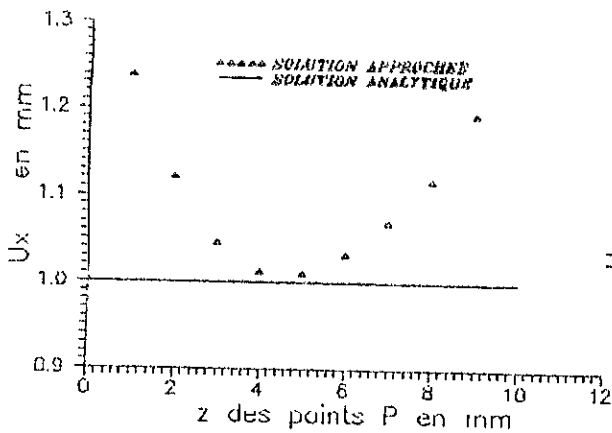
Deplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'un déplacement du corps rigide
 $U_x = 1$ $U_y = 0$ $U_z = 0$ en mm
 Maillage de la surface par 24 éléments

Deplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'un déplacement du corps rigide
 $U_x = 1$ $U_y = 0$ $U_z = 0$ en mm
 Maillage de la surface par 24 éléments

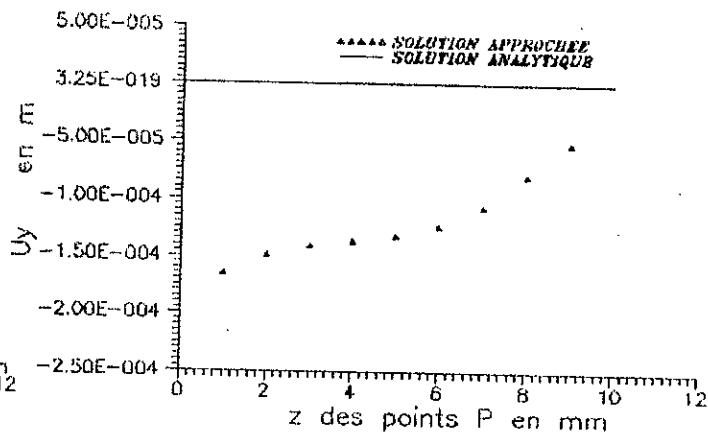


Deplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'un déplacement du corps rigide
 $U_x = 1$ $U_y = 0$ $U_z = 0$ en mm
 Maillage de la surface par 24 éléments

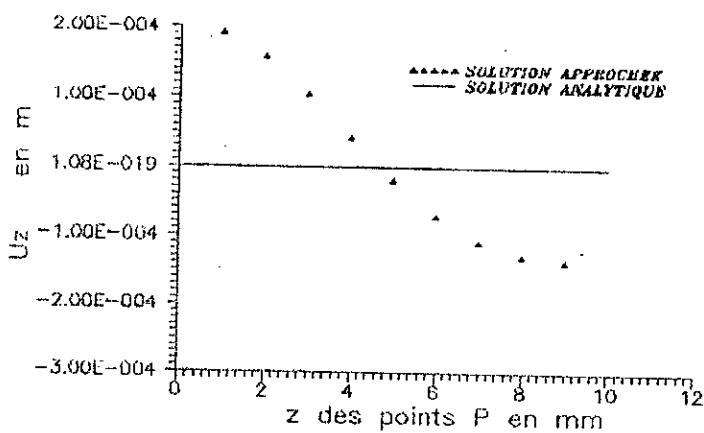
Fig 5.5



Déplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'un déplacement du corps rigide
 $U_x = 1$ $U_y = 0$ $U_z = 0$ en mm
 Maillage de la surface par 54 éléments



Déplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'un déplacement du corps rigide
 $U_x = 1$ $U_y = 0$ $U_z = 0$ en mm
 Maillage de la surface par 54 éléments



Déplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'un déplacement du corps rigide
 $U_x = 1$ $U_y = 0$ $U_z = 0$ en mm
 Maillage de la surface par 54 éléments

Fig 5.6

On remarque que :

- Les résultats numériques des points proches du centre du cube sont plus précis : l'erreur est de l'ordre de 1% , alors que les points proches de la surface ont une erreur qui atteint 25% .

Ce resultat confirme les résultats obtenus par S.L.CROUCH [3] dans le cas bidimensionnel : Il a remarqué que les résultats divergent pour les points proches du contour d'une distance égale à la taille de l'élément .

- Les contraintes en ces points sont très loins de la solution analytique : la solution analytique pour le cas d'un déplacement du corps rigide est :

$$\sigma_{ij} = 0 \quad i, j = 1, 2, 3$$

Mais le résultat numérique donne des contraintes de l'ordre de 10^9 Pa .

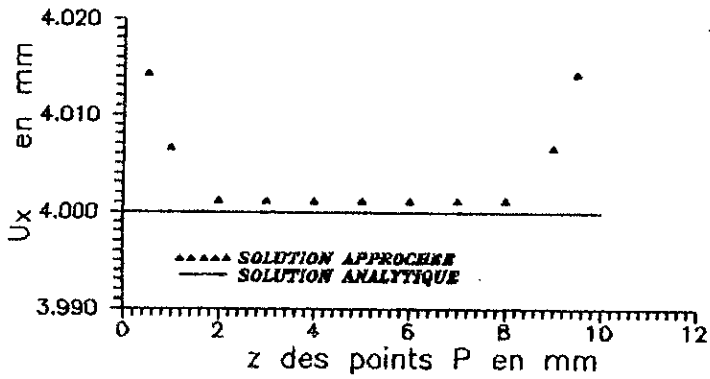
Pour expliquer ce résultat on a étudié un deuxième exemple .

EXEMPLES 2 : on prend dans cet exemple les conditions aux limites suivantes :

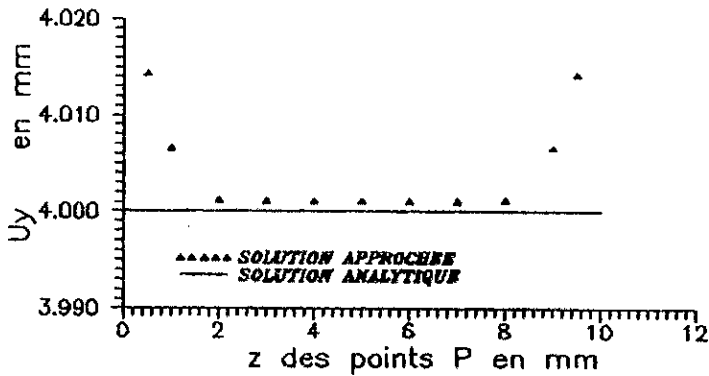
$$U_x = U_y = U_z = 4 \text{ mm} .$$

Pour éliminer les erreurs numériques dues à la première étape de calcul (calcul des tensions sur la surface) on prend directement la solution analytique : $T_x = T_y = T_z = 0$ sur la surface et on passe à la seconde étape : calcul des déplacements et des contraintes des points intérieurs .

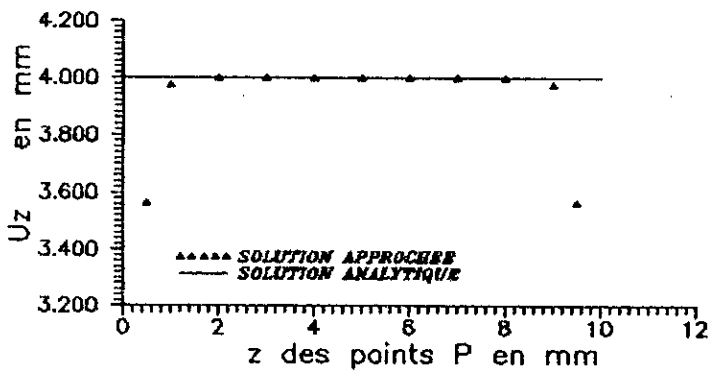
La figure 5.7 représente les composantes du vecteur déplacement des points P de la droite ($x = 5$, $y = 5$) en fonction de z , la surface étant discrétisée en 54 éléments .



Déplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'un déplacement du corps rigide
 $U_x = 4$ $U_y = 4$ $U_z = 4$ en mm
 Maillage de la surface par 54 éléments



Déplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'un déplacement du corps rigide
 $U_x = 4$ $U_y = 4$ $U_z = 4$ en mm
 Maillage de la surface par 54 éléments



Déplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'un déplacement du corps rigide
 $U_x = 4$ $U_y = 4$ $U_z = 4$ en mm
 Maillage de la surface par 54 éléments

Fig 5.7

On remarque que :

- Comme précédemment , les résultats sont plus précis pour les points proches du centre du cube .

- Les déplacements sont symétriques par-rapport au plan $z = 5$

- $U_x = U_y \neq U_z$. Ce qui est normale puisque les points P sont indifférents par-rapport aux axes x et y seulement .

Les figures 5.8 , 5.9 et 5.10 représentent les composantes du tenseur des contraintes des points P de la droite ($x = 5$, $y = 5$) en fonction de z . La surface est discrétisée en 54 éléments .

On constate que :

- $\sigma_{ij} = 0 \quad \forall i,j$: Le centre du cube n'est soumis à aucune contrainte. On peut expliquer ce résultat par le fait que le champ des déplacements trouvé numériquement est symétrique par-rapport à ce point, donc les déformations "numériques" sont nulles en ce point et par conséquent les contraintes sont nulles aussi .

-Si on s'éloigne du centre du cube dans la direction des z , les contraintes σ_{11} , σ_{22} , σ_{33} , σ_{13} et σ_{23} augmentent considérablement . Cela peut être expliqué par le fait que le champ des déplacements trouvé numériquement n'est pas constant , et puisqu'il y a variation des déplacements alors il y a déformations non nulles et par conséquent des contraintes non nulles .

Si on fait un calcul approximatif de la contrainte à partir des déplacements on aura pour le point P($z = 1$ mm) :

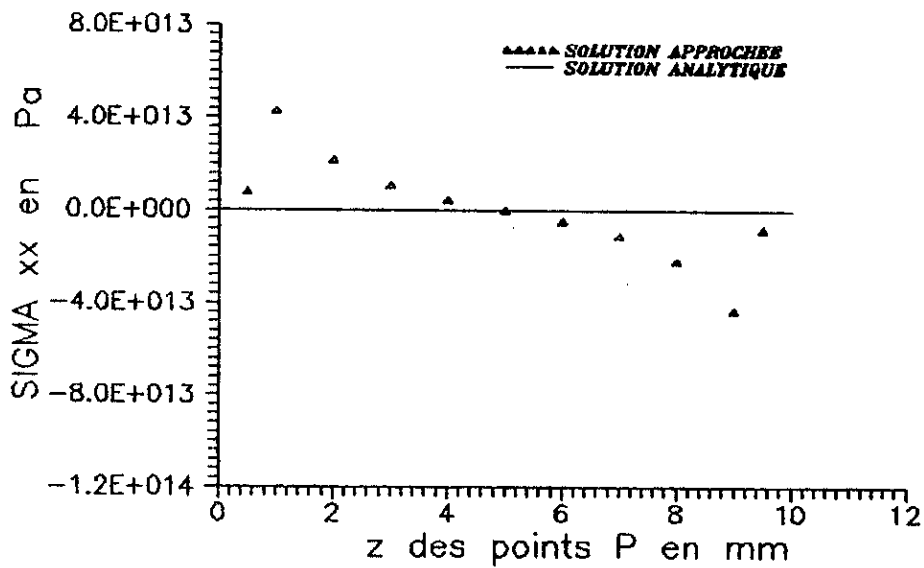
$$\varepsilon_{11} = \frac{U_1(z=1) - U_1(z=0.5)}{1 - 0.5} = \frac{4.014 - 4.0065}{0.5}$$

$$\varepsilon_{11} = 0.015$$

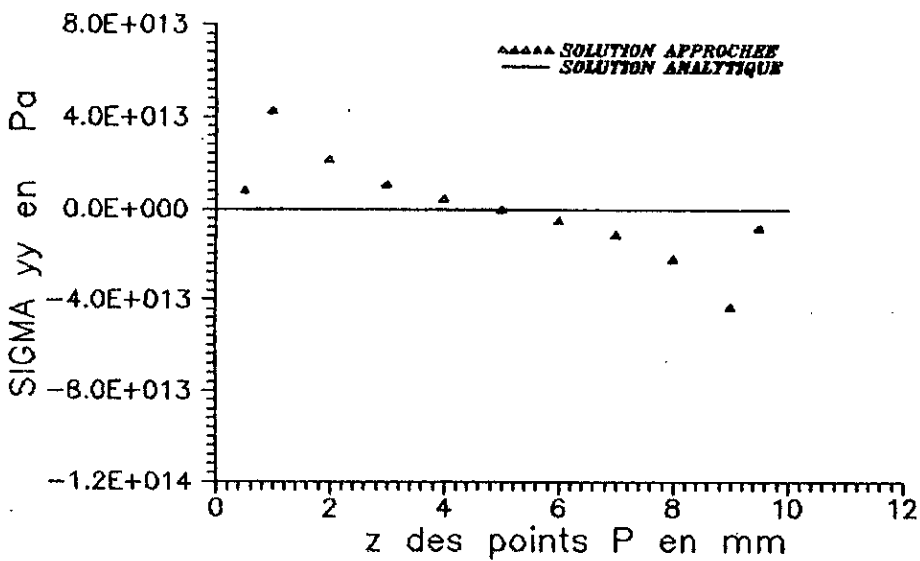
$\nu = 0$ d'ou :

$$\sigma_{11} = E \varepsilon_{11} = 200 \cdot 10^9 \cdot 0.015 = 3 \cdot 10^9 \text{ Pa}$$

ce qui nous donne le même ordre de grandeur pour les contraintes trouvées numériquement .

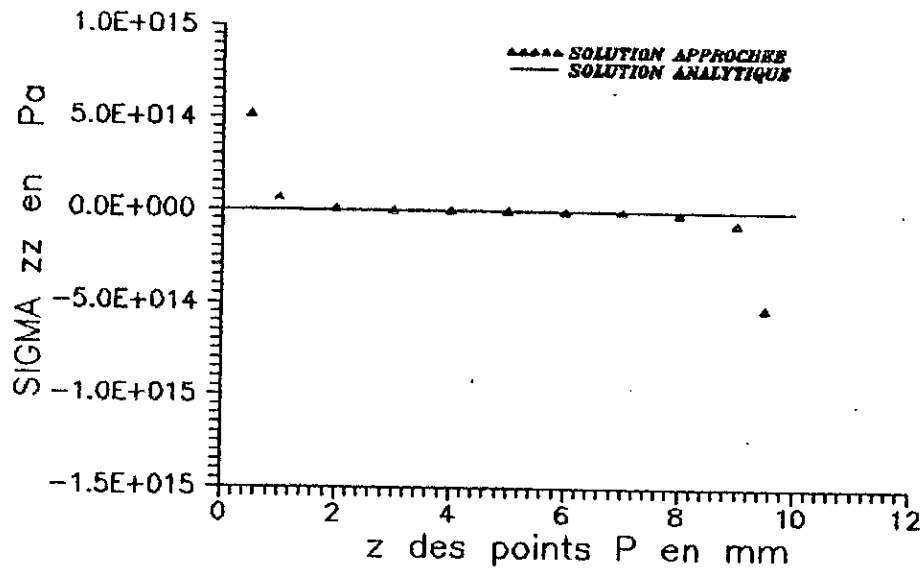


contraintes des points de la droite $x = 5$, $y = 5$
 Pour le cas d'un déplacement du corps rigide
 $U_x = 4$, $U_y = 4$, $U_z = 4$ en mm
 Maillage de la surface par 54 elements

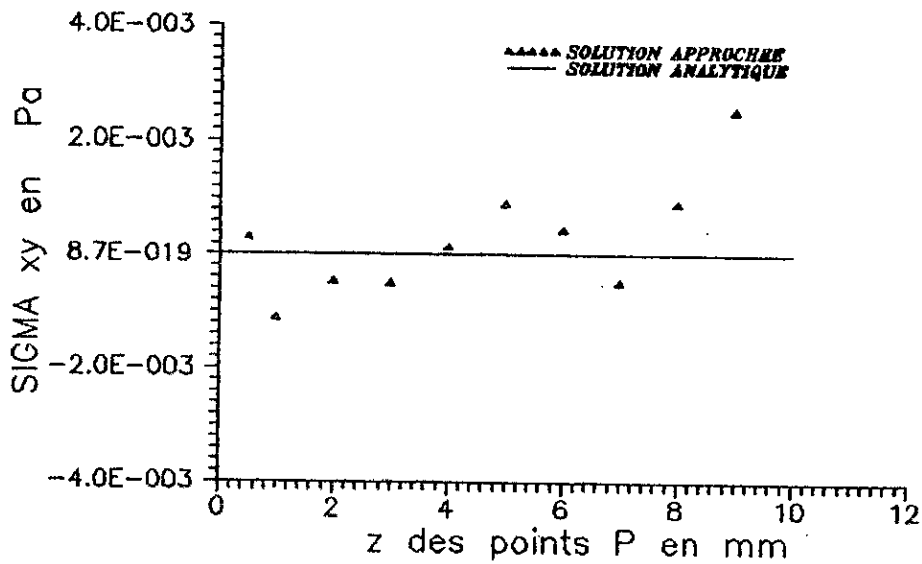


contraintes des points de la droite $x = 5$, $y = 5$
 Pour le cas d'un déplacement du corps rigide
 $U_x = 4$, $U_y = 4$, $U_z = 4$ en mm
 Maillage de la surface par 54 elements

Fig 5.8

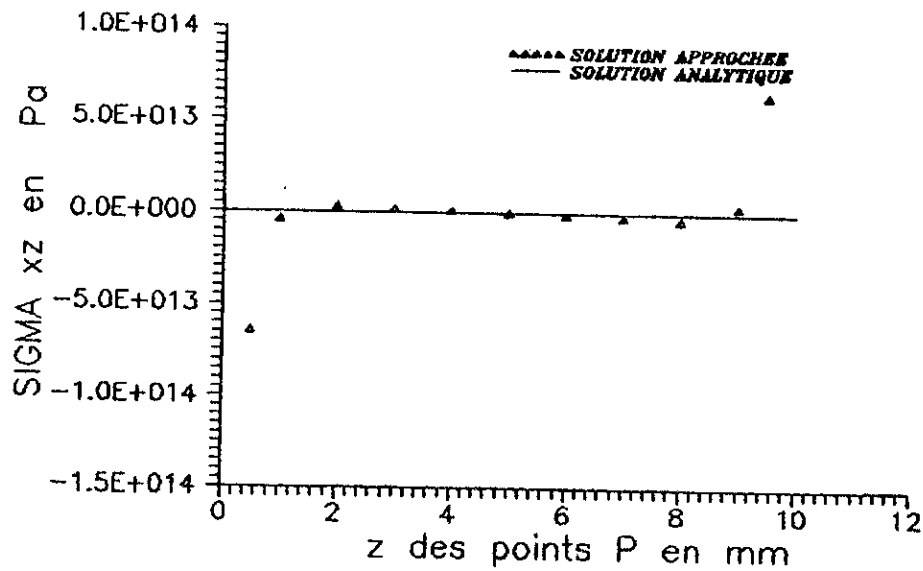


contraintes des points de la droite $x = 5$, $y = 5$
 Pour le cas d'un déplacement du corps rigide
 $U_x = 4$, $U_y = 4$, $U_z = 4$ en mm
 Maillage de la surface par 54 elements

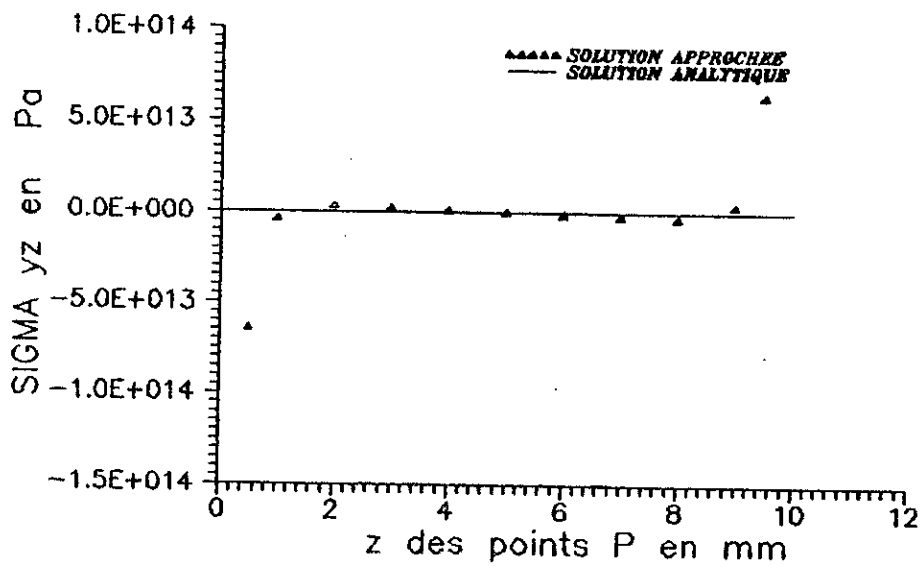


contraintes des points de la droite $x = 5$, $y = 5$
 Pour le cas d'un déplacement du corps rigide
 $U_x = 4$, $U_y = 4$, $U_z = 4$ en mm
 Maillage de la surface par 54 elements

Fig 5.9



contraintes des points de la droite $x = 6$, $y = 5$
 Pour le cas d'un déplacement du corps rigide
 $U_x = 4$, $U_y = 4$, $U_z = 4$ en mm
 Maillage de la surface par 54 éléments



contraintes des points de la droite $x = 5$, $y = 5$
 Pour le cas d'un déplacement du corps rigide
 $U_x = 4$, $U_y = 4$, $U_z = 4$ en mm
 Maillage de la surface par 54 éléments

Fig 5.10

Pour obtenir une bonne approximation des contraintes il est nécessaire de raffiner le maillage de la surface .

- On constate que σ_{12} est pratiquement nulle pour tout les points $P(z)$. Cela s'explique par le fait que la droite $(x = 5 , y = 5)$ est un axe de symetrie pour champ des déplacements donc ϵ_{12} est nulle et par conséquent σ_{12} l'est aussi .

5.2 CUBE EN TRACTION :

EXEMPLE 1 :

Le cube est soumis à une charge T_z selon l'axe z .

Les figures 5.11 , 5.12 , 5.13 et 5.14 représentent les composantes du vecteur déplacements pour des valeurs du coefficient de POISSON égales respectivement à 0 , 0.24 , 0.35 , 0.45 .

La valeur de la charge $T_z = 10^{10}$ Pa

$E = 200 \cdot 10^9$ Pa

La surface est discrétisée en 24 éléments .

Les conditions aux limites sont :

$T_z = 10^{10}$ Pa pour les éléments 13,14,15 et 16 .

$U_z = 0$ pour les éléments 5,6,7 et 8 .

$U_x = U_y = 0$ pour l'élément 5

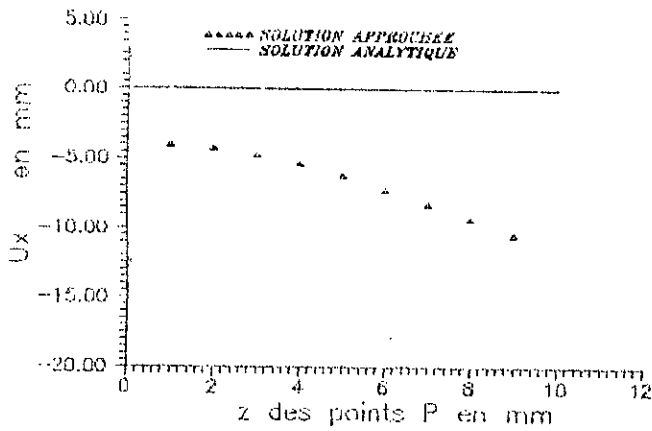
$U_x = 0$ pour l'élément 7.

On constate que :

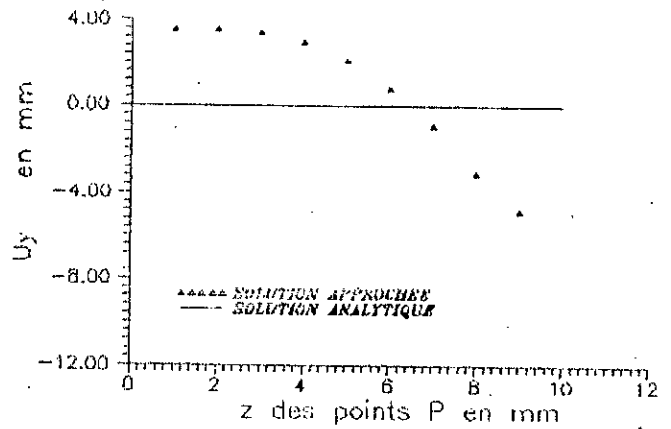
- Les composantes U_x et U_y sont plus précises pour les points proches du centre du cube

- $U_x(P) \neq U_y(P)$ car les conditions aux limites de déplacements ne sont pas symétriques .

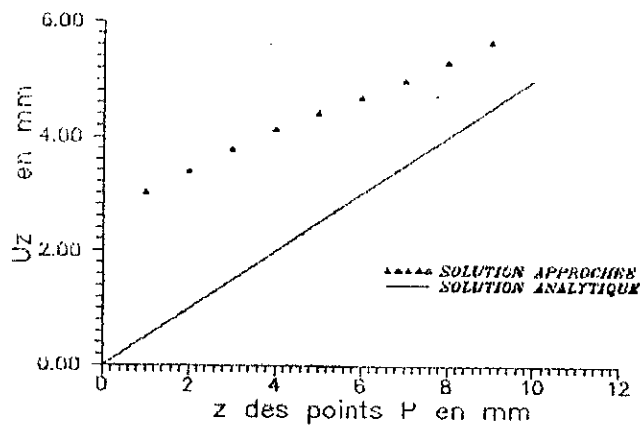
- Les résultats sont moins précises par-rapport à l'exemple précédent car Le maillage est grossier (24 éléments)



Déplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'une traction uniforme suivant z
 $T_x = 10 \text{ GPa/m}^2$ $E = 200 \text{ GPa/m}^2$ $\nu = 0$
 Maillage de la surface par 24 éléments

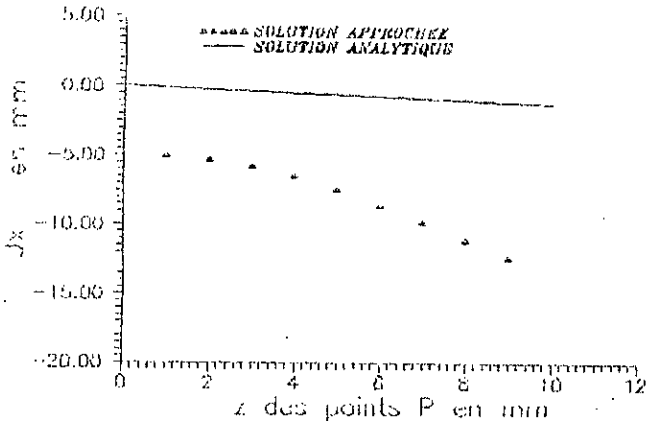


Déplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'une traction uniforme suivant z
 $T_x = 10 \text{ GPa/m}^2$ $E = 200 \text{ GPa/m}^2$ $\nu = 0$
 Maillage de la surface par 24 éléments

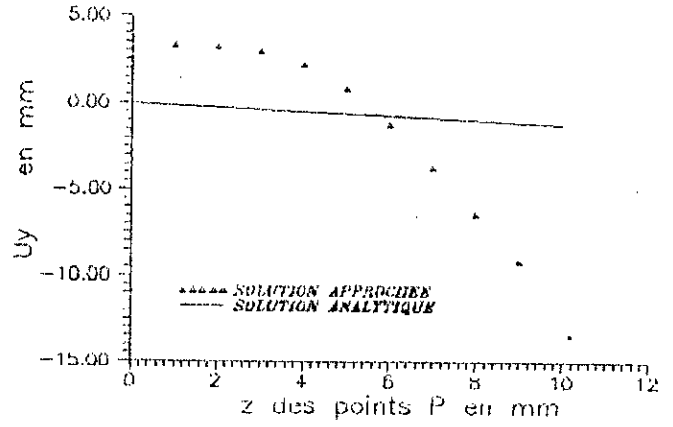


Déplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'une traction uniforme suivant z
 $T_x = 10 \text{ GPa/m}^2$ $E = 200 \text{ GPa/m}^2$ $\nu = 0$
 Maillage de la surface par 24 éléments

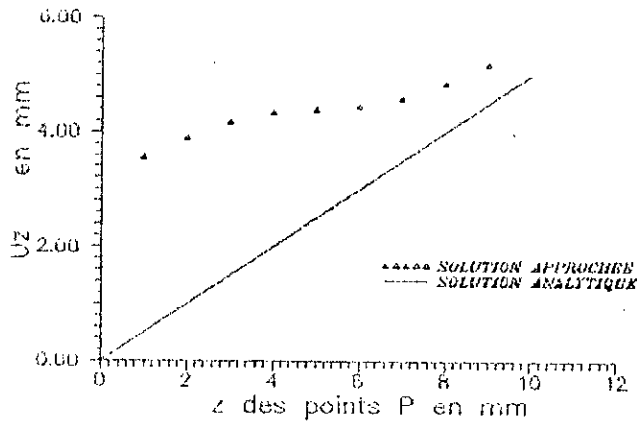
Fig 5.11



Déplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'une traction uniforme suivant z
 $T_z = 10 \text{ GPa/m}^2$ $E = 200 \text{ GPa/m}^2$ $\nu = 0.24$
 Maillage de la surface par 24 éléments

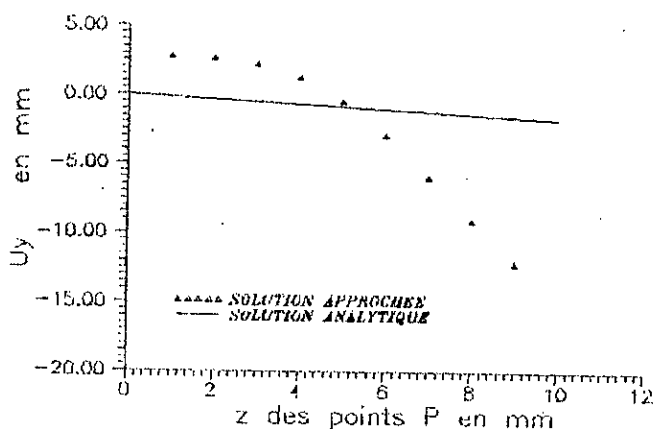
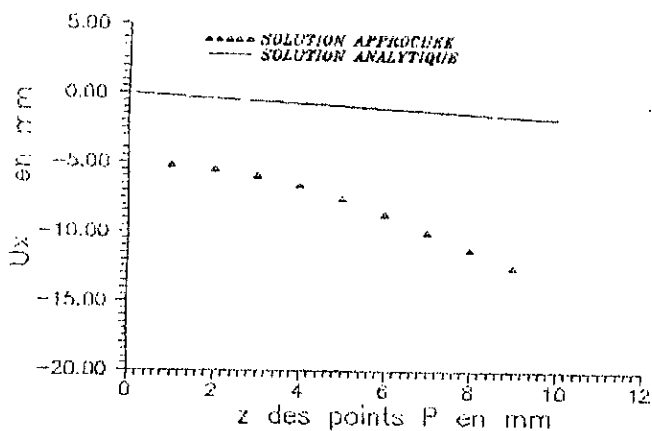


Déplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'une traction uniforme suivant z
 $T_z = 10 \text{ GPa/m}^2$ $E = 200 \text{ GPa/m}^2$ $\nu = 0.24$
 Maillage de la surface par 24 éléments



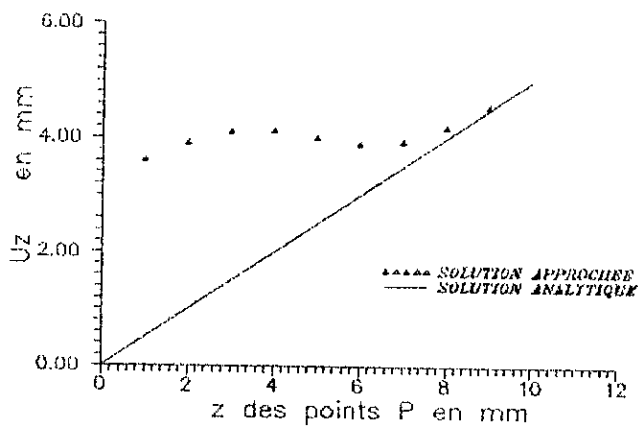
Déplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'une traction uniforme suivant z
 $T_z = 10 \text{ GPa/m}^2$ $E = 200 \text{ GPa/m}^2$ $\nu = 0.24$
 Maillage de la surface par 24 éléments

Fig 5.12



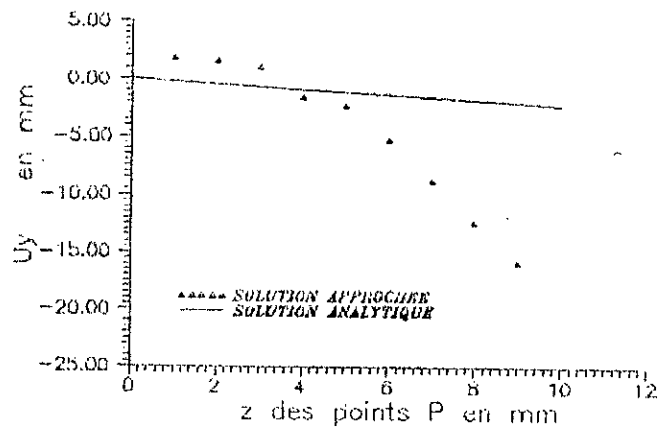
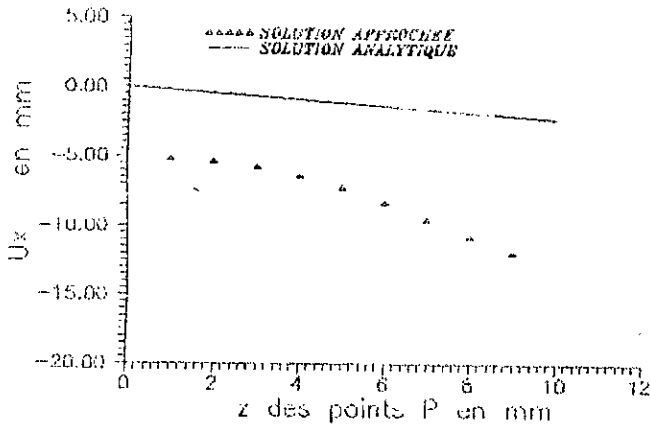
Déplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'une traction uniforme suivant z
 $T_z = 10 \text{ GPa/m}^2$ $E = 200 \text{ GPa/m}^2$ $\nu = 0.35$
 Maillage de la surface par 24 éléments

Déplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'une traction uniforme suivant z
 $T_z = 10 \text{ GPa/m}^2$ $E = 200 \text{ GPa/m}^2$ $\nu = 0.35$
 Maillage de la surface par 24 éléments



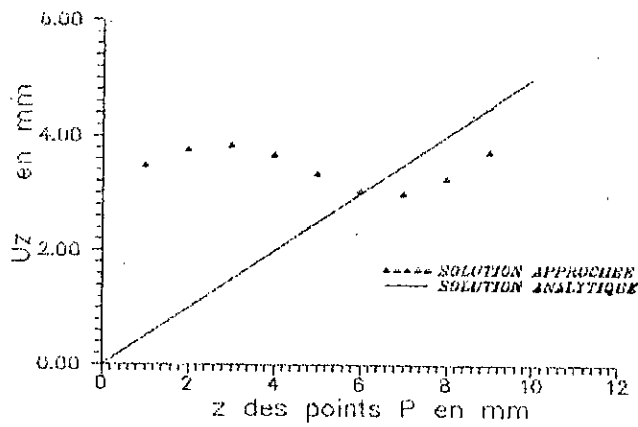
Déplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'une traction uniforme suivant z
 $T_z = 10 \text{ GPa/m}^2$ $E = 200 \text{ GPa/m}^2$ $\nu = 0.35$
 Maillage de la surface par 24 éléments

Fig 5.13



Déplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'une traction uniforme suivant z
 $T_z = 10 \text{ CPa/m}^2$ $E = 200 \text{ GPa/m}^2$ $\nu = 0.45$
 Maillage de la surface par 24 éléments

Déplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'une traction uniforme suivant z
 $T_z = 10 \text{ CPa/m}^2$ $E = 200 \text{ GPa/m}^2$ $\nu = 0.45$
 Maillage de la surface par 24 éléments



Déplacement des points de la droite $x = 5$ $y = 5$
 pour le cas d'une traction uniforme suivant z
 $T_z = 10 \text{ CPa/m}^2$ $E = 200 \text{ GPa/m}^2$ $\nu = 0.45$
 Maillage de la surface par 24 éléments

Fig 5.14

EXMPLE 2 :

Le cube est soumis à une traction $T_z = 0.95 \cdot 10^{10}$ Pa .

$E = 200 \cdot 10^9$ Pa

Le maillage de la surface est fait par 54 éléments .

Les figures 5.15 , 5.16 et 5.17 représentent les composantes du tenseur des contraintes des points de la droite ($x = 5, y = 5$).

On remarque que :

- Les résultats les plus précis sont pour les points loins de la surface .

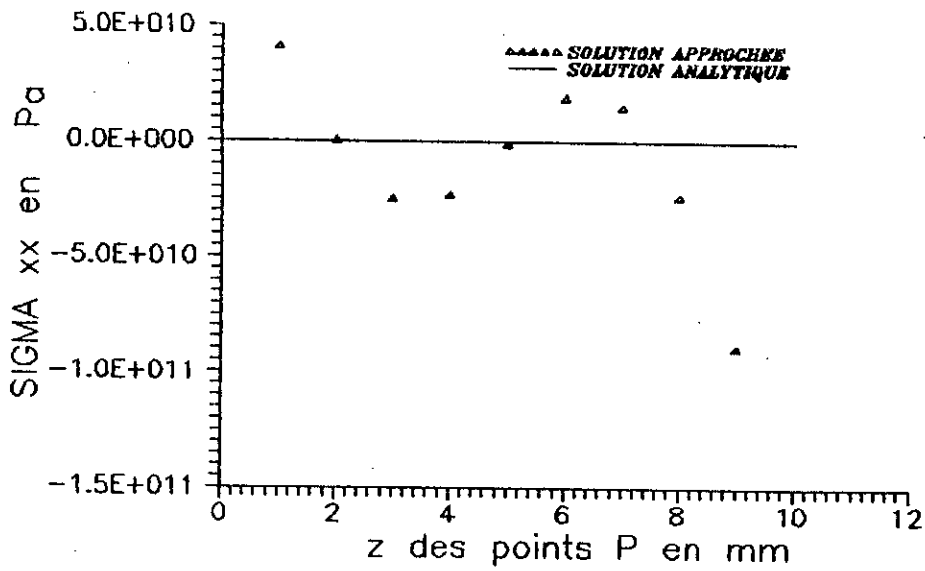
- $\sigma_{11}(P) \neq \sigma_{22}(P)$ car les conditions aux limites ne sont pas les mêmes selon les axes x et y .

- L'erreur comise sur les contraintes est grande , cela est du à :

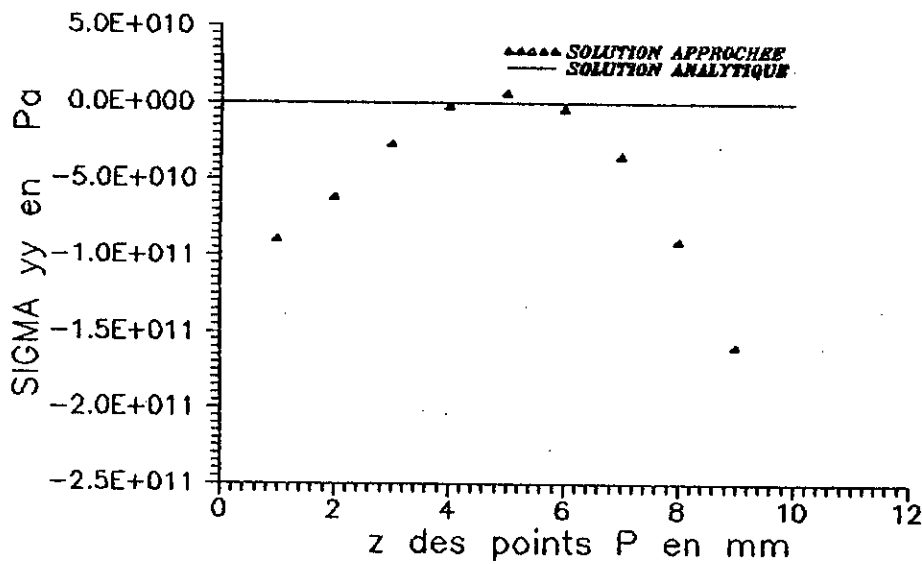
1- Le maillage n'est pas fin :il faut noter que Pour les problèmes bidimensionnels simples ,il a fallut 200 éléments pour obtenir une précision de 3% (S.L.CROUCH [5]).donc il ne faut pas espérer avoir une bonne précision avec 54 éléments pour un problème tridimensionnel.

pour pouvoir utiliser un grand nombre d'éléments , il est nécessaire de prendre en compte la symetrie du problème ,ce qui permet dans plusieurs cas de raffiner quatre fois de plus .

2- Le fait de supposer que U_z est constant pour les éléments latéraux cela induit une grande erreur .L'utilisation des éléments d'ordres supérieurs est nécessaire.

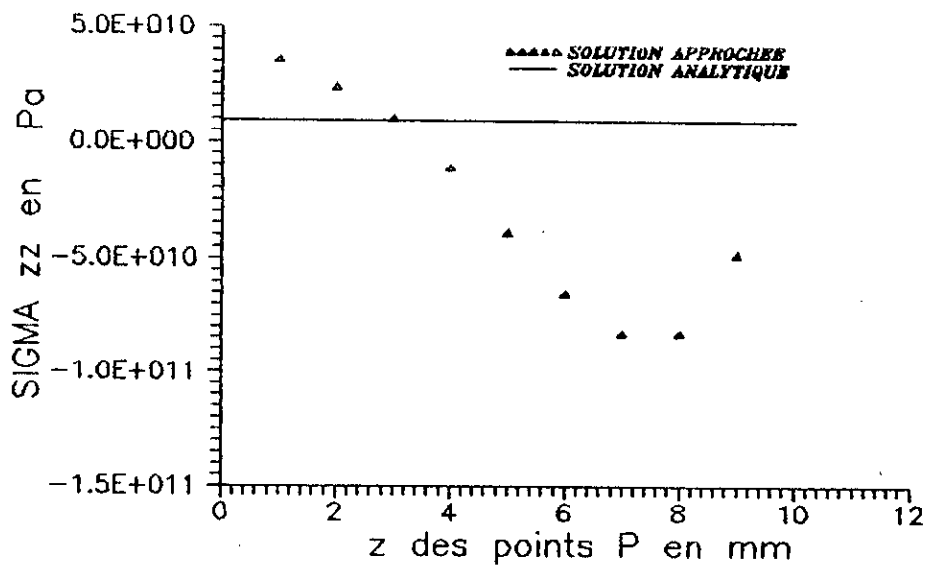


contraintes des points de la droite $x = 5$, $y = 5$
 Pour le cas d'une traction uniforme suivant z
 $T_z = 0.95$ Gega Pa $E = 200$ Gega Pa $\nu = 0.24$
 Maillage de la surface par 54 elements

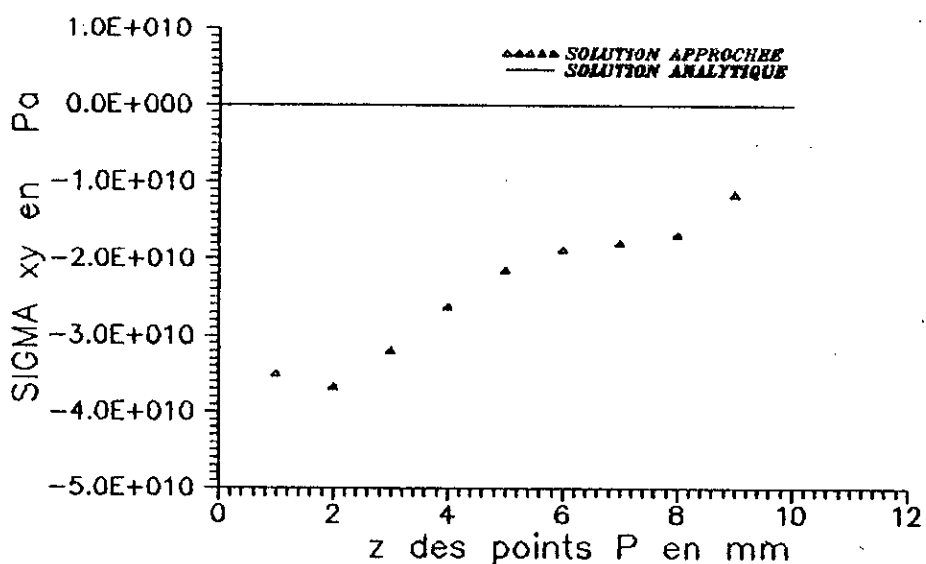


contraintes des points de la droite $x = 5$, $y = 5$
 Pour le cas d'une traction uniforme suivant z
 $T_z = 0.95$ Gega Pa $E = 200$ Gega Pa $\nu = 0.24$
 Maillage de la surface par 54 elements

Fig 5.15

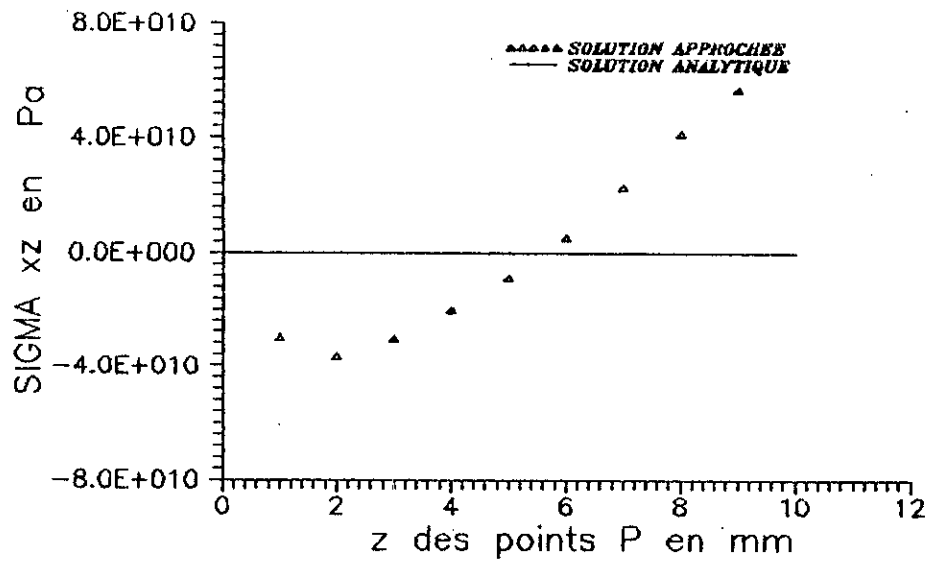


contraintes des points de la droite $x = 5$, $y = 5$
 Pour le cas d'une traction uniforme suivant z
 $T_z = 0.95$ Gega Pa $E = 200$ Gega Pa $\nu = 0.24$
 Maillage de la surface par 54 elements

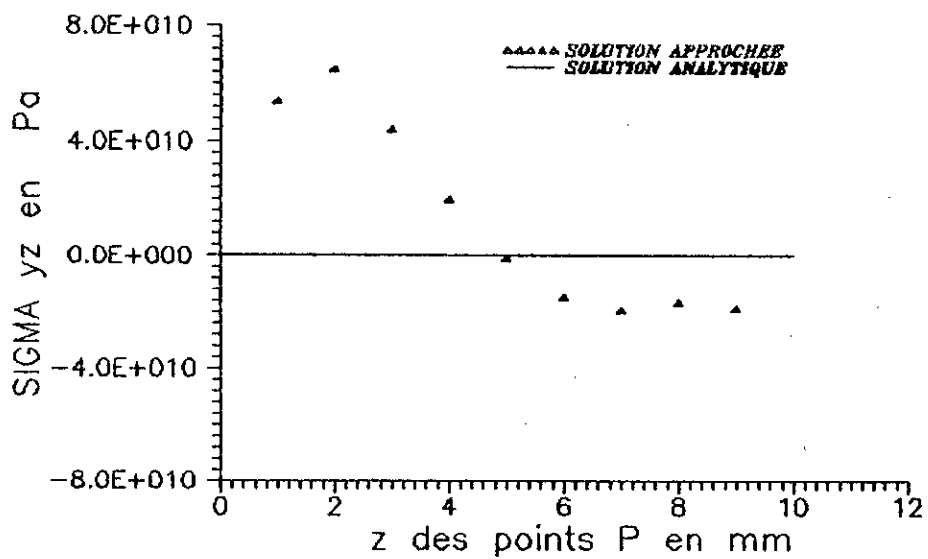


contraintes des points de la droite $x = 5$, $y = 5$
 Pour le cas d'une traction uniforme suivant z
 $T_z = 0.95$ Gega Pa $E = 200$ Gega Pa $\nu = 0.24$
 Maillage de la surface par 54 elements

Fig 5.16



contraintes des points de la droite $x = 5$, $y = 5$
 Pour le cas d'une traction uniforme suivant z
 $T_z = 0.95$ Gega Pa $E = 200$ Gega Pa $\nu = 0.24$
 Maillage de la surface par 54 elements



contraintes des points de la droite $x = 5$, $y = 5$
 Pour le cas d'une traction uniforme suivant z
 $T_z = 0.95$ Gega Pa $E = 200$ Gega Pa $\nu = 0.24$
 Maillage de la surface par 54 elements

Fig 5.17

CONCLUSION

CONCLUSION

On est arrivé à travers cette étude à élaborer un programme utilisant la méthode des équations intégrale en trois dimensions, qui nous permet de déterminer le champ des déplacements et des contraintes pour un corps homogène, élastique et linéaire .

Les résultats numériques étaient plus satisfaisants pour le champ des déplacements que pour le champ des contraintes .

Je pense que le but a été atteint , il reste à développer ce programme pour le rendre plus puissant , pour cela je suggère de traiter ultérieurement les points suivants :

- L'utilisation des éléments d'ordre supérieur .
- Tenir compte de la symétrie du problème posé .

Ce qui permet de résoudre efficacement des problèmes plus complexes .

En fin , je tiens à signaler que plusieurs autres problèmes n'ont pas été présentés à cause de la non disponibilité du matériel informatique adéquat .

ANNEXE

LISTING DU PROGRAMME M.E.I.T.D.

```

c  ****
c  *                               *
c  *           METHODE INTEGRALE DIRECTE           *
c  *           EN TROIS DIMENSIONS                 *
c  * Détermination du champ des déplacements et des contraintes *
c  *           dans un milieu homogène élastique   *
c  *                               *
c  * PROPOSE PAR :M.BELKACEMI           ETUDIE PAR : KEBIR HOCINE *
c  ****

c  ****PRINC
c  *                               *PRINC
c  ****PRINC
  implicit double precision (a-h,o-z) PRINC
  common/keb1/ d(162,6,3) PRINC
  common/keb2/a(162,162) PRINC
  common/keb3/b(162,162) PRINC
  common/keb4/c1(162,2,3) PRINC
  common/keb5/cpdi(150,5,4) PRINC
  common/keb6/cp(300,3) PRINC
  common/keb7/u(162) PRINC
  common/keb8/t(162) PRINC
  common/keb9/f(162) PRINC
  common/keb10/ki(162) PRINC
  common/keb11/ppi(100,3) PRINC
  common/keb12/ppid(100,3) PRINC
  common/keb13/ppic(100,3,3) PRINC
c  -----PRINC
c  LECTURE DES FICHER PRINC
c  -----PRINC
  character*20 donnee PRINC
  character*20 result PRINC
7542 format(a20) PRINC
  WRITE(*,*) 'DONNER LE FICHER DE DONNEES' PRINC
  READ(*,7542)DONNEE PRINC
  WRITE(*,*) 'DONNER LE FICHER DE RESULTATS' PRINC
  READ(*,7542)RESULT PRINC
  open (unit=1,file=donnee,status='unknown') PRINC
  open (unit=2,file=result,status='unknown') PRINC
c  ****PRINC
  call lecture(nplan,E,xNU,ndepl,nchar,npi) PRINC
  call mailleur(nplan,nde) PRINC
  call AetB(nde,e,xnu,g) PRINC
  call KetF(nchar,ndepl,nde,n,g) PRINC

```

```

call resLU(n) PRINC
call rearg(n,ndepl,g) PRINC
call Dpint(npi,nde,e,xnu) PRINC
call cpint(npi,nde,e,xNU) PRINC
call affich(nde,npi) PRINC
c *****PRINC
write(*,*)'les resultats sont dans le fichier : ', result PRINC
stop PRINC
end PRINC
c *****PRINC

c *****LECTUR
c * SUBROUTINE : LECTURE DES DONNEES *LECTUR
c *****LECTUR
subroutine lecture(nplan,E,xNU,ndepl,nchar,npi) LECTUR
implicit double precision (a-h,o-z) LECTUR
common/keb5/cpdi(150,5,4) LECTUR
common/keb6/cp(300,3) LECTUR
common/keb4/cl(162,2,3) LECTUR
common/keb11/ppi(100,3) LECTUR
read(1,*)nn LECTUR
read(1,*)nplan LECTUR
do i=1,nn LECTUR
read(1,*)(cp(i,j),j=1,3) LECTUR
end do LECTUR
do i=1,nplan LECTUR
read(1,*)(cpdi(i,j,4),j=1,4) LECTUR
read(1,*)cpdi(i,5,1),cpdi(i,5,2) LECTUR
end do LECTUR
do i=1,nplan LECTUR
do j=1,4 LECTUR
n=cpdi(i,j,4) LECTUR
do 15 k=1,3 LECTUR
15 cpdi(i,j,k)=cp(n,k) LECTUR
enddo LECTUR
enddo LECTUR
read(1,*) E,xNU LECTUR
c ----- LECTUR
read(1,*)ndepl LECTUR
do i=1,ndepl LECTUR
read(i,*)(cl(i,1,k),k=1,3) LECTUR

```

```

end do
read(1,*)nchar
do i=1,nchar
read(1,*)(c1(i,2,k),k=1,3)
end do
c
-----
read(1,*)npi
do i=1,npi
read(1,*)(ppi(i,j),j=1,3)
end do
return
end
c
*****
c
*****
c
SUBROUTINE : LE MAILLEUR
c
*****
c
*****
c
*   entrée: la lecture se fait a l'interieur de la subroutine *
c
*   nn : nombre de noeuds qui definissent les plans *
c
*   nplan : nombre de plan *
c
*   cp(i,j) : pour chaque noeuds i      i=1,nn *
c
*               cp(i,j) est la Jeme coordonnée *
c
*               *
c
*   cpd(i,j,k) : pour chaque plan i      i=1,nplan *
c
*               j=1,4 *
c
*   cpd(i,j,4) est le numeros du Jeme eleant *
c
*               *
c
*               *
c
*****
c
subroutine mailleur(nplan,nde)
implicit double precision (a-h,o-z)
common/keb6/cp(300,3)
common/keb5/cpdi(150,5,4)
common/keb1/d(162,6,3)
dimension kelm(100,4),cpd(4,4)
dimension sol(5),cpd7(4,3),w1(3),w2(3),w3(3)
c
-----
c
EXECUTION
c
-----
c
nde=0
ndp=0

```

	do i=1,nplan	MAILS
	nl=cpdi(i,5,1)	MAILS
	nc=cpdi(i,5,2)	MAILS
	do 4 j=1,4	MAILS
	do 4 k=1,4	MAILS
4	cpd(j,k)=cpdi(i,j,k)	MAILS
	cpd(1,4)=ndp+1	MAILS
	call mailplan(nde,nl,nc,ndp,keln,cpd)	MAILS
	end do	MAILS
	do 13 i=1,nde	MAILS
	do 13 j=1,4	MAILS
	do 13 k=1,3	MAILS
	ll=keln(i,j)	MAILS
	d(i,j,k)=cp(ll,k)	MAILS
13	continue	MAILS
c	-----	MAILS
c	calcul des coordonnées du milieu de l'element	MAILS
c	-----	MAILS
	do 14 i=1,nde	MAILS
	do k=1,3	MAILS
	cpd7(1,k)=(d(i,2,k)+d(i,1,k))/2.	MAILS
	cpd7(2,k)=(d(i,4,k)+d(i,3,k))/2.	MAILS
	cpd7(3,k)=(d(i,3,k)+d(i,1,k))/2.	MAILS
	cpd7(4,k)=(d(i,4,k)+d(i,2,k))/2.	MAILS
	end do	MAILS
	call intersec(cpd7,sol)	MAILS
	do k=1,3	MAILS
	d(i,6,k)=sol(k)	MAILS
	end do	MAILS
c	-----	MAILS
c	calcul de la normale à cet element	MAILS
c	-----	MAILS
	do k=1,3	MAILS
	w1(k)=d(i,3,k)-d(i,1,k)	MAILS
	w2(k)=d(i,2,k)-d(i,1,k)	MAILS
	end do	MAILS
	call prodvec(w1,w2,w3)	MAILS
	do k=1,3	MAILS
	d(i,5,k)=w3(k)	MAILS
	end do	MAILS
14	continue	MAILS


```

c      _____ MAILS
c      AFFICHAGE DES RESULTAS MAILS
c      _____ MAILS
do ii=1,nde MAILS
write(2,*)'numeros de l'element : ',ii MAILS
do j=1,6 MAILS
write(2,*)(d(ii,j,k),k=1,3) MAILS
end do MAILS
write(2,*) MAILS
end do MAILS
c      _____ MAILS
return MAILS
end MAILS
c      *****MAILS

c      *****MAILP
c      SUBROUTINE : MAILPLAN MAILP
c      *****MAILP
subroutine mailplan(nde,nl,nc,ndp,kelm,cpd) MAILP
implicit double precision (a-h,o-z) MAILP
common /keb6/ cp(300,3) MAILP
dimension kelm(100,4),cpd(4,4),cpd7(4,3),sol(5) MAILP
n1=cpd(1,4) MAILP
n2=n1+n1 MAILP
n3=n1+nc*(n1+1) MAILP
n4=n3+n1 MAILP
do j=1,3 MAILP
cp(n1,j)=cpd(1,j) MAILP
cp(n2,j)=cpd(2,j) MAILP
cp(n3,j)=cpd(3,j) MAILP
cp(n4,j)=cpd(4,j) MAILP
end do MAILP
do i=1,nl-1 MAILP
do j=1,3 MAILP
cp(n1+i,j)=cp(n1,j)+i*(cp(n2,j)-cp(n1,j))/nl MAILP
cp(n3+i,j)=cp(n3,j)+i*(cp(n4,j)-cp(n3,j))/nl MAILP
end do MAILP
end do MAILP
do i=1,nc-1 MAILP
do j=1,3 MAILP
cp(n1+(nl+1)*i,j)=cp(n1,j)+i*(cp(n3,j)-cp(n1,j))/nc MAILP

```

	end do	MATLP
	end do	MATLP
	l11=(nl+1)*(nc+1)	MATLP
	do 10 i=1,l11	MATLP
	nci=(i-1)/(nl+1)+1	MATLP
	li=i-(nci-1)*(nl+1)	MATLP
	nk=nl+1	MATLP
	if((li.eq.1).or.(li.eq.nk)) go to 10	MATLP
	nf=nc+1	MATLP
	if((nci.eq.1).or.(nci.eq.nf)) go to 10	MATLP
	m1=li-1+nl	MATLP
	m2=m1+nc*(nl+1)	MATLP
	m3=n1+(nci-1)*(nl+1)	MATLP
	m4=m3+nl	MATLP
	na1=m4	MATLP
	do j=1,3	MATLP
	cpd7(1,j)=cp(m1,j)	MATLP
	cpd7(2,j)=cp(m2,j)	MATLP
	cpd7(3,j)=cp(m3,j)	MATLP
	cpd7(4,j)=cp(m4,j)	MATLP
	end do	MATLP
	call intersec(cpd7,sol)	MATLP
	do ii=1,3	MATLP
	cp(i+nl-1,ii)=sol(ii)	MATLP
	end do	MATLP
10	continue	MATLP
	do j=1,nl*nc	MATLP
	icj=(j-1)/nl+1	MATLP
	lj=j-(nl*(icj-1))	MATLP
	m1=(nl+1)*(icj-1)+lj+nl-1	MATLP
	m2=m1+1	MATLP
	m3=m1+nl+1	MATLP
	m4=m3+1	MATLP
	num=nde+j	MATLP
	keln(num,1)=m1	MATLP
	keln(num,2)=m2	MATLP
	keln(num,3)=m3	MATLP
	keln(num,4)=m4	MATLP
	end do	MATLP
	ndp=ndp+(nl+1)*(nc+1)	MATLP
	nde=num	MATLP
	return	MATLP

```

end
c *****MAILP
c *****
c          SUBROUTINE QUI FORME LES MATRICES [A] ET [B]
c *****
subroutine AetB(nde,E,xNU,g)
implicit double precision (a-h,o-z)
common/keb1/d(162,6,3)
common/keb2/a(162,162)
common/keb3/b(162,162)
dimension cpd5(5,3),cpd3(2,3)
dimension p(3,4),p1(3,3),v(3),a1o(3),cpd6(5,3),a1(3),a2(3),a3(3)
dimension a4(3)
data eps/1.d-14/
data pi/3.141592653589793/

n=nde*3
do ii=1,n
xii=ii
xin=n
dp=(xii/xin)*100
ip=dp
write(*,2030)ip
2030 format(40(/),20x,i4,'% du travail est executé',8(/))
write(*,2040)
2040 format(2(/),25x,'S.V.P NE TOUCHER PAS ',/,30x,'MERCI')
do jj=1,n
ly=(jj-1)/3+1
ldy=jj-(3*(ly-1))
lx=(ii-1)/3+1
ldx=ii-(3*(lx-1))
do 10 i=1,5
do 10 j=1,3
10 cpd5(i,j)=d(ly,i,j)
do 20 i=1,3
cpd3(1,i)=d(lx,6,i)
20 cpd3(2,i)=0.
if(lx.eq.ly) then
npint=8
call matpass(cpd5,p)
call inverse(p,p1)

```

do k=1,3	AetB
v(k)=cpd3(1,k)-cpd5(1,k)	AetB
end do	AetB
call prodmv(p1,3,v,a1o)	AetB
a1(1)=a1o(1)-eps	AetB
a1(2)=a1o(2)-eps	AetB
a1(3)=0.	AetB
a2(1)=a1o(1)-eps	AetB
a2(2)=a1o(2)+eps	AetB
a2(3)=0.	AetB
a3(1)=a1o(1)+eps	AetB
a3(2)=a1o(2)-eps	AetB
a3(3)=0.	AetB
a4(1)=a1o(1)+eps	AetB
a4(2)=a1o(2)+eps	AetB
a4(3)=0.	AetB
call prodmv(p,3,a1,v)	AetB
do k=1,3	AetB
a1(k)=cpd5(1,k)+v(k)	AetB
end do	AetB
call prodmv(p,3,a2,v)	AetB
do k=1,3	AetB
a2(k)=cpd5(1,k)+v(k)	AetB
end do	AetB
call prodmv(p,3,a3,v)	AetB
do k=1,3	AetB
a3(k)=cpd5(1,k)+v(k)	AetB
end do	AetB
call prodmv(p,3,a4,v)	AetB
do k=1,3	AetB
a4(k)=cpd5(1,k)+v(k)	AetB
end do	AetB
do k=1,3	AetB
cpd6(5,k)=cpd5(5,k)	AetB
end do	AetB
do k=1,3	AetB
cpd6(3,k)=cpd5(1,k)	AetB
cpd6(1,k)=cpd5(2,k)	AetB
cpd6(4,k)=a1(k)	AetB
cpd6(2,k)=a2(k)	AetB
end do	AetB
call integra(cpd6,cpd3,ldx,ldy,xNU,npint,T1,U1,0)	AetB

```

do k=1,3
cpd6(1,k)=cpd5(1,k)
cpd6(2,k)=a1(k)
cpd6(3,k)=cpd5(3,k)
cpd6(4,k)=a3(k)
end do
call integra(cpd6,cpd3,ldx,ldy,xNU,npint,T2,U2,0)
do k=1,3
cpd6(2,k)=a3(k)
cpd6(4,k)=a4(k)
cpd6(1,k)=cpd5(3,k)
cpd6(3,k)=cpd5(4,k)
end do
call integra(cpd6,cpd3,ldx,ldy,xNU,npint,T3,U3,0)
do k=1,3
cpd6(4,k)=a2(k)
cpd6(3,k)=cpd5(2,k)
cpd6(2,k)=a4(k)
cpd6(1,k)=cpd5(4,k)
end do
call integra(cpd6,cpd3,ldx,ldy,xNU,npint,T4,U4,0)
t=t1+t2+t3+t4
u=u1+u2+u3+u4
a(ii,jj)=t+4.*pi*(1.-xnu)*chron(ldy,ldx)
b(ii,jj)=u
else
npint=6
call integra(cpd5,cpd3,ldx,ldy,xNU,npint,T,U,0)
a(ii,jj)=t
b(ii,jj)=u
end if
end do
end do
g=e/(2.*(1.+xnu))
return
end
c *****AetB
c ***** KetF
c SUBROUTINE QUI FORME LA MATRICE [K] ET LE VECTEUR {F} KetF
c ***** KetF
subroutine KetF(nchar,ndepl,nde,n,g) KetF

```

```

implicit double precision (a-h,o-z)
common/keb4/cl(162,2,3)
common/keb2/a(162,162)
common/keb3/b(162,162)
common/keb8/t(162)
common/keb7/u(162)
common/keb9/f(162)
common/keb10/ki(162)
dimension f1(162)
n=nde*3
do i=1,n
f1(i)=0.
t(i)=0.
end do
do i=1,nchar
ll=(cl(i,2,2)-1)*3+cl(i,2,3)
f1(ll)=cl(i,2,1)/(2.*g)
t(ll)=cl(i,2,1)
end do
do i=1,ndepl
ll=(cl(i,1,2)-1)*3+cl(i,1,3)
ki(i)=ll
f1(ll)=cl(i,1,1)
u(ll)=cl(i,1,1)
do j=1,n
temp=a(j,ll)
a(j,ll)=b(j,ll)
b(j,ll)=temp
end do
end do
call prodmv(b,n,f1,f)
return
end
c *****
C *****RESLU
C      SOUS-PROGRAMME DE FACTORISATION DE LA MATRICE A      RESLU
C              A = L * U      RESLU
C *****RESLU
SUBROUTINE RESLU(N)      RESLU
IMPLICIT DOUBLEPRECISION (A-H,O-Z)      RESLU
common/keb2/a(162,162)      RESLU

```



```

END DO RESLU
C ----- RESLU
RETURN RESLU
900 write(2,*) 'LE PIVOT DE CETTE LIGNE EST NUL' RESLU
write(*,*) 'le pivot de cette ligne est nul' RESLU
stop RESLU
END RESLU
c *****RESLU

c *****REARG
c * SUBROUTINE REARRANGEMENT DES SOLUTIONS *REARG
c *****REARG
subroutine rearg(n,ndepl,g) REARG
implicit double precision (a-h,o-z) REARG
common/keb10/ki(162) REARG
common/keb8/t(162) REARG
common/keb9/f(162) REARG
common/keb7/u(162) REARG
do ii=1,n REARG
n=0 REARG
do j=1,ndepl REARG
if(ii.eq.ki(j)) then REARG
n=1 REARG
end if REARG
end do REARG
if(n.eq.1) then REARG
t(ii)=f(ii)*2.*g REARG
else REARG
u(ii)=f(ii) REARG
end if REARG
end do REARG
return REARG
end REARG
c *****REARG

c *****DPINT
c SUBROUTINE QUI CALCUL LES DEPLACEMENTS DPINT
c DES POINTS INTERIEURS DPINT
c *****DPINT
subroutine Dpint(npi,nde,e,xnu) DPINT
implicit double precision (a-h,o-z) DPINT
common/keb7/u(162) DPINT

```


common/keb8/t(162)	DPINT
common/keb1/d(162,6,3)	DPINT
common/keb11/ppi(100,3)	DPINT
common/keb12/ppid(100,3)	DPINT
dimension cpd3(2,3),cpd5(5,3)	DPINT
data npint/6/	DPINT
data pi/3.141592653589793/	DPINT
n=nde*3	DPINT
do i=1,np1	DPINT
do j=1,3	DPINT
cpd3(1,j)=ppi(i,j)	DPINT
end do	DPINT
do ii=1,3	DPINT
ppid(i,ii)=0.	DPINT
do j=1,n	DPINT
ly=((j-1)/3)+1	DPINT
ldy=j-(3*(ly-1))	DPINT
do i10=1,5	DPINT
do j10=1,3	DPINT
cpd5(i10,j10)=d(ly,i10,j10)	DPINT
end do	DPINT
end do	DPINT
call integra(cpd5,cpd3,ii,ldy,xnu,npint,tt,uu,0)	DPINT
tt=tt*(1./(8.*pi**((1.-xnu))))	DPINT
uu=uu*(1.+xnu)/(8.*pi*e*(1.-xnu))	DPINT
ppid(i,ii)=ppid(i,ii)-tt*u(j)+uu*t(j)	DPINT
end do	DPINT
end do	DPINT
end do	DPINT
return	DPINT
end	DPINT
c *****DPINT	
c *****CPINT	
c SUBROUTINE QUI CALCUL LES CONTRAINTES	CPINT
c DES POINTS INTERIEURS	CPINT
c *****CPINT	
subroutine Cpint(np1,nde,R,xNU)	CPINT
implicit double precision (a-h,o-z)	CPINT
common/keb7/u(162)	CPINT
common/keb8/t(162)	CPINT
common/keb1/d(162,6,3)	CPINT

```

common/keb11/ppi(100,3) CPINT
common/keb13/ppic(100,3,3) CPINT
dimension cpd3(2,3),cpd5(5,3) CPINT
data pi/3.141592653589793/ CPINT
data npint/6/ CPINT
n=nde*3 CPINT
do i=1,np1 CPINT
do j=1,3 CPINT
cpd3(1,j)=ppi(i,j) CPINT
end do CPINT
do ii=1,3 CPINT
do jj=1,3 CPINT
ppic(i,ii,jj)=0. CPINT
do j=1,n CPINT
ly=((j-1)/3)+1 CPINT
ldy=j-(3*(ly-1)) CPINT
do i10=1,5 CPINT
do j10=1,3 CPINT
cpd5(i10,j10)=d(ly,i10,j10) CPINT
end do CPINT
end do CPINT
call integra(cpd5,cpd3,ii,jj,xnu,npint,tt,uu,ldy) CPINT
ppic(i,ii,jj)=ppic(i,ii,jj)+uu*t(j)-tt*E*u(j) CPINT
end do CPINT
ppic(i,ii,jj)=ppic(i,ii,jj)/(8.*pi*(1.-xnu)) CPINT
end do CPINT
end do CPINT
end do CPINT
return CPINT
end CPINT
c *****CPINT
c *****AFFICH
c AFFICHAGE DES RESULTATS AFFICH
c *****AFFICH
subroutine AFFICH(nde,np1) AFFICH
implicit double precision (a-h,o-z) AFFICH
common/keb7/u(162) AFFICH
common/keb8/t(162) AFFICH
common/keb11/ppi(100,3) AFFICH
common/keb12/ppid(100,3) AFFICH
common/keb13/ppic(100,3,3) AFFICH

```

character*2 so	AFICH
n=nde*3	AFICH
write(2,1001)	AFICH
do i=1,n	AFICH
ll=i-((i-1)/3)*3	AFICH
nnoeud=(i-1)/3+1	AFICH
if(ll.eq.1)then	AFICH
so='Ux'	AFICH
else	AFICH
if(ll.eq.2)then	AFICH
so='Uy'	AFICH
else	AFICH
so='Uz'	AFICH
end if	AFICH
end if	AFICH
write(2,1003)so,nnoeud,u(i)	AFICH
end do	AFICH
write(2,1002)	AFICH
do i=1,n	AFICH
ll=i-((i-1)/3)*3	AFICH
nnoeud=(i-1)/3+1	AFICH
if(ll.eq.1)then	AFICH
so='Tx'	AFICH
else	AFICH
if(ll.eq.2)then	AFICH
so='Ty'	AFICH
else	AFICH
so='Tz'	AFICH
end if	AFICH
end if	AFICH
write(2,1003)so,nnoeud,t(i)	AFICH
end do	AFICH
write(2,*)'LES DEPLACEMENTS DES POINTS INTERIEURS'	AFICH
do i=1,npi	AFICH
write(2,*)(ppi(i,j),j=1,3),(ppid(i,j),j=1,3)	AFICH
end do	AFICH
write(2,*)	AFICH
write(2,*)'LES CONTRAINTES DES POINTS INTERIEURS'	AFICH
do i=1,npi	AFICH
write(2,*)	AFICH
write(2,*)'numero du point : ',i	AFICH
write(2,*)(ppi(i,j),j=1,3)	AFICH

```

write(2,*)
do j=1,3
write(2,*)(ppic(i,j,k),k=1,3)
end do
write(2,*)
end do
1001 format('LE CHAMP DES DEPLACEMENTS EST :',//)
1002 format('//, 'LE CHAMP DES CONTRAINTES EST :',//)
1003 format(4x,a2,'(',i2,') = ',f36.16)
return
end
c *****AFICH
c *****INTEG
c          SUBROUTINE QUI CALCUL LES VALEURS DES INTEGRALES          INTEG
c *****INTEG
c *****INTEG
c *   ENTREE :   cpd5(5,3) :          *INTEG
c *               pour chaque noeud i ,i=1,4          *INTEG
c *               cpd5(i,j) est la Jeme coordonnees          *INTEG
c *               cpd5(5,j) coordonnées de la normale          *INTEG
c *               cpd3(2,3) :          *INTEG
c *               cpd3(1,j) coordonnées de X          *INTEG
c *               cpd3(2,j) sera definit a l'interieur          *INTEG
c *               ldx : direction sur l'element X          *INTEG
c *               ldy : direction sur l'element Y          *INTEG
c *               E :module de young          *INTEG
c *               xNU :   w          *INTEG
c *               npint : nombre de point de gauss          *INTEG
c *   SORTIE :   T valeur de l'integrale du noyau Tij          *INTEG
c *               U valeur de l'integrale du noyau Uij          *INTEG
c *****INTEG
c
c          subroutine integra(cpd5,cpd3,ldx,ldy,xNU,npint,T,U,k)          INTEG
c          implicit double precision (a-h,o-z)          INTEG
c          dimension cpd5(5,3),cpd3(2,3),p(3,4),p1(3,3),coorp4(3)          INTEG
c          dimension v(3),xn(3),coorpg(10,10),coefg(10,10)          INTEG
c
c          coordonnes des points de gauss          INTEG
c
c          data coorpg/0.d0, 0.d0 , 0.d0 , 0.d0 , 0.d0 , 0.d0, 0.d0, 0.d0, INTEG
+ 0.d0, 0.d0,          INTEG

```

+ -.5773502691896d0, .5773502691896d0 , 0.d0 , 0.d0,	INTEG
+ 0.d0, 0.d0, 0.d0, 0.d0, 0.d0, 0.d0,	INTEG
+ -.7745966692415d0,0.d0, .7745966692415d0 , 0.d0 ,	INTEG
+ 0.d0, 0.d0, 0.d0, 0.d0, 0.d0, 0.d0,	INTEG
+ -.8611363115941d0, -.3399810435849d0 ,	INTEG
+ .3399810435849d0 , .8611363115941d0,0.d0,0.d0,0.d0,	INTEG
+ 0.d0, 0.d0, 0.d0,	INTEG
+ -.9061798459d0 ,-.5384693101d0 ,0.d0 ,.5384693101d0,	INTEG
+ .9061798459d0, 0.d0, 0.d0, 0.d0, 0.d0, 0.d0,	INTEG
+ -.9324695142d0, -.6612093865d0,-.2386191861d0,	INTEG
+ .2386191861d0, .6612093865d0, .8324695142d0,0.d0,0.d0,	INTEG
+ 0.d0,0.d0,	INTEG
+ -.9491079123d0 ,-.7415311856d0, -.4058451514d0,0.d0,	INTEG
+ .4058451514d0,.7415311856d0, .9491079123d0 , 0.d0,0.d0,	INTEG
+ 0.d0,	INTEG
+ -.9602898565d0, -.7966664774d0, -.5255324099d0,	INTEG
+ -.1834346425d0, .1834346425d0, .5255324099d0, .7966664774d0,	INTEG
+ .9602898565d0,0.d0,0.d0,	INTEG
+ -.9681602395d0, -.8360311073d0,-.6133714327d0,	INTEG
+ -.3242534234d0,0.d0, .3242534234d0, .6133714327d0,	INTEG
+ .8360311073d0, .9681602395d0,0.d0,	INTEG
+ -.9739065285d0, -.8650633667d0, -.6794095683d0,	INTEG
+ -.4333953941d0, -.1488743390d0, .1488743390d0, .4333953941d0,	INTEG
+ .6794095683d0, .8650633667d0, .9739065285d0/	INTEG
c	INTEG
coefficient de gauss : poids de gauss	INTEG
	INTEG
data coefg/2.d0,0.d0,0.d0,0.d0,0.d0,0.d0,0.d0,0.d0,0.d0,0.d0,	INTEG
+ 1.d0,1.d0,0.d0,0.d0,0.d0,0.d0,0.d0,0.d0,0.d0,0.d0,	INTEG
+ .5555555555556d0, .8888888888889d0, .5555555555556d0,0.d0,0.d0,	INTEG
+ 0.d0,0.d0,0.d0,0.d0,0.d0,	INTEG
+ .3478548451375d0, .6521451548625d0, .6521451548625d0,	INTEG
+ .3478548451375d0,0.d0,0.d0,0.d0,0.d0,0.d0,0.d0,	INTEG
+ .2369268850d0, .4786286705d0, .56888888889d0, .4786286705d0,	INTEG
+ .2369268850d0,0.d0,0.d0,0.d0,0.d0,0.d0,	INTEG
+ .1713244924d0, .3607615730d0, .4679139346d0, .4679139346d0,	INTEG
+ .3607615730d0, .1713244924d0,0.d0, 0.d0, 0.d0, 0.d0,	INTEG
+ .1294849662d0, .2797053915d0, .3818300505d0, .4179591837d0,	INTEG
+ .3818300505d0, .2797053915d0, .1294849662d0,0.d0, 0.d0,0.d0,	INTEG
+ .1012285363d0, .2223810345d0, .3137066459d0, .3626837834d0,	INTEG
+ .3626837834d0, .3137066459d0, .2223810345d0, .1012285363d0,	INTEG
+ 0.d0,0.d0,	INTEG

+ .0812743884d0, .1806481607d0, .2606106964d0, .3123470770d0,	INTEG
+ .3302393550d0, .3123470770d0, .2606106964d0, .1806481607d0,	INTEG
+ .0812743884d0,0.d0,	INTEG
+ .0666713443d0, .1494513492d0, .2190863625d0, .2692667193d0,	INTEG
+ .2955242247d0, .2955242247d0, .2692667193d0, .2190863625d0,	INTEG
+ .1494513492d0, .0666713443d0/	INTEG
 	INTEG
do i=1,3	INTEG
xn(i)=cpd5(5,i)	INTEG
end do	INTEG
call matpass(cpd5,p)	INTEG
call inverse(p,p1)	INTEG
do i=1,3	INTEG
v(i)=cpd5(4,i)-cpd5(1,i)	INTEG
end do	INTEG
call prodmv(p1,3,v,coop4)	INTEG
x1=coop4(1)	INTEG
y1=coop4(2)	INTEG
do i=1,3	INTEG
v(i)=cpd5(3,i)-cpd5(1,i)	INTEG
end do	INTEG
call prodmv(p1,3,v,coop4)	INTEG
x4=coop4(1)	INTEG
y4=coop4(2)	INTEG
do i=1,3	INTEG
v(i)=cpd5(2,i)-cpd5(1,i)	INTEG
end do	INTEG
call prodmv(p1,3,v,coop4)	INTEG
x2=coop4(1)	INTEG
y2=coop4(2)	INTEG
x3=0.	INTEG
y3=0.	INTEG
T=0.	INTEG
U=0.	INTEG
do lx=1,npint	INTEG
r=cooprg(lx,npint)	INTEG
do ly=1,npint	INTEG
s=cooprg(ly,npint)	INTEG
x=(1./4.)*((1.+r)*(1.+s)*x1+(1.-r)*(1.+s)*x2+(1.-r)*(1.-s)*x3+(1.	INTEG
+r)*(1.-s)*x4)	INTEG
y=(1./4.)*((1.+r)*(1.+s)*y1+(1.-r)*(1.+s)*y2+(1.-r)*(1.-s)*y3+(1.	INTEG
+r)*(1.-s)*y4)	INTEG

```

xJ11=(1./4.)*((1.+s)*x1-(1.+s)*x2-(1.-s)*x3+(1.-s)*x4)      INTEG
xJ21=(1./4.)*((1.+r)*x1+(1.-r)*x2-(1.-r)*x3-(1.+r)*x4)      INTEG
xJ12=(1./4.)*((1.+s)*y1-(1.+s)*y2-(1.-s)*y3+(1.-s)*y4)      INTEG
xJ22=(1./4.)*((1.+r)*y1+(1.-r)*y2-(1.-r)*y3-(1.+r)*y4)      INTEG
detJ=xj11*xj22-xj21*xj12                                       INTEG
coop4(1)=x                                                         INTEG
coop4(2)=y                                                         INTEG
coop4(3)=0.                                                       INTEG
call prodmv(p,3,coop4,v)                                          INTEG
do 30 i=1,3                                                       INTEG
30 cpd3(2,i)=cpd5(1,i)+v(i)                                       INTEG
   if(k.eq.0) then                                               INTEG
     call noyaut(ldx,ldy,cpd3,xn,xNU,Tij)                         INTEG
     call noyauU(ldx,ldy,cpd3,xNU,Uij)                           INTEG
   else                                                           INTEG
     call noyaus(ldx,ldy,k,cpd3,xn,xnu,tij)                       INTEG
     call noyaud(ldx,ldy,k,cpd3,xnu,u ij)                         INTEG
   end if                                                         INTEG
xinaam=coefg(lx,npint)*coefg(ly,npint)*detJ                       INTEG
T=T+xinaam*Tij                                                    INTEG
U=U+xinaam*Uij                                                    INTEG
end do                                                            INTEG
end do                                                            INTEG
return                                                            INTEG
end                                                                INTEG
c *****INTEG

c *****NOYT
e   SUBROUTINE : LE NOYAU Tij(x,y)                                NOYT
c *****NOYT
subroutine noyaut(ldx,ldy,cpd3,xn,xNU,Tij)                        NOYT
implicit double precision (a-h,o-z)                              NOYT
dimension cpd3(2,3),xn(3)                                        NOYT
call dis(cpd3,r)                                                NOYT
d=chron(ldx,ldy)                                                NOYT
t1=1./(r**2)                                                    NOYT
t2=xn(ldx)*(cpd3(1,ldy)-cpd3(2,ldy))/r                          NOYT
t3=xn(ldy)*(cpd3(1,ldx)-cpd3(2,ldx))/r                          NOYT
t4=(1.-2.*xNU)*d                                               NOYT
t5=3.*(cpd3(1,ldx)-cpd3(2,ldx))*(cpd3(1,ldy)-cpd3(2,ldy))    NOYT
t6=0                                                            NOYT

```

```

do i=1,3
t6=t6+xn(i)*(cpd3(1,i)-cpd3(2,i))
end do
t6=t6/r
Tij=t1*((1.-2.*xNU)*(t2-t3)+(t4+t5/(r**2))*t6)
return
end
c *****NOYT

c *****NOYU
c      SUBROUTINE : LE NOYAU Uij(x,y)
c *****NOYU
subroutine noyau(ldx,ldy,cpd3,xNU,Uij)
implicit double precision (a-h,o-z)
dimension cpd3(2,3)
call dis(cpd3,r)
d=chron(ldx,ldy)
Uij=(1./r)*((3.-4.*xNU)*d+((cpd3(1,ldx)-cpd3(2,ldx))*(cpd3(1,ld
+y)-cpd3(2,ldy)))/r**2)
return
end
c *****NOYU

c *****NOYD
c      SUBROUTINE : LE NOYAU Dijk(x,y)
c *****NOYD
subroutine noyand (i,j,k,cpd3,xNU,Dijk)
implicit double precision (a-h,o-z)
dimension cpd3(2,3)
call dis(cpd3,r)
d1=chron(i,j)
d2=chron(i,k)
d3=chron(j,k)
t1=1./r**2
t2=d1*(cpd3(1,k)-cpd3(2,k))/r
t3=d2*(cpd3(1,j)-cpd3(2,j))/r
t4=d3*(cpd3(1,i)-cpd3(2,i))/r
t5=3.*(cpd3(1,i)-cpd3(2,i))*(cpd3(1,j)-cpd3(2,j))
t5=t5*(cpd3(1,k)-cpd3(2,k))/(r**3)
dijk=t1*(((1.-2.*xnu)*(t2-t3-t4))-t5)
return
c *****NOYD

```



```

end NOYD
c *****NOYD
c *****NOYS
c SUBROUTINE : LE NOYAU Sijk(x,y) NOYS
c *****NOYS
subroutine noyaus(i,j,k,cpd3,xn,xnu,Sijk) NOYS
implicit double precision (a-h,o-z) NOYS
dimension cpd3(2,3),xn(3) NOYS
call dis(cpd3,r) NOYS
d1=chron(i,j) NOYS
d2=chron(i,k) NOYS
d3=chron(j,k) NOYS
t1=1./((1.+xnu)*r**3) NOYS
t2=0. NOYS
do l=1,3 NOYS
t2=t2+xn(l)*(cpd3(1,l)-cpd3(2,l)) NOYS
end do NOYS
t2=t2/r NOYS
t31=3.*(1.-2.*xnu)*d1*(cpd3(1,k)-cpd3(2,k))/r NOYS
t32=3.*xnu*(d2*(cpd3(1,j)-cpd3(2,j))+d3*(cpd3(1,i)-cpd3(2,i))) NOYS
t32=t32/r NOYS
t33=15.*(cpd3(1,i)-cpd3(2,i))*(cpd3(1,j)-cpd3(2,j)) NOYS
t33=t33*(cpd3(1,k)-cpd3(2,k))/r**3 NOYS
t3=t31+t32-t33 NOYS
t41=3.*xnu*(cpd3(1,j)-cpd3(2,j))*(cpd3(1,k)-cpd3(2,k))/r**2 NOYS
t42=(1.-2.*xnu)*d3 NOYS
t4=xn(i)*(t41+t42) NOYS
t51=3.*xnu*(cpd3(1,i)-cpd3(2,i))*(cpd3(1,k)-cpd3(2,k))/r**2 NOYS
t52=(1.-2.*xnu)*d2 NOYS
t5=xn(j)*(t51+t52) NOYS
t61=3.*xnu*(cpd3(1,i)-cpd3(2,i))*(cpd3(1,j)-cpd3(2,j))/r**2 NOYS
t62=(1.-2.*xnu)*d1 NOYS
t6=xn(k)*(t61+t62) NOYS
Sijk=t1*(t2*t3+t4+t5+t6) NOYS
return NOYS
end NOYS
c *****NOYS
c *****INTER
c SUBROUTINE : COORDONNEES DU POINT D'INTERSECTION INTER
c *****INTER

```

```

subroutine intersec(cpd7,sol)                                INTERS
implicit double precision (a-h,o-z)                        INTERS
dimension cpd7(4,3),sol(5),coor(5,6)                      INTERS
do 111 i5=1,5                                              INTERS
do111 j=1,5                                                INTERS
coor(i5,j)=0                                               INTERS
111  continue                                             INTERS
do i5=1,3                                                  INTERS
coor(i5,i5)=1                                             INTERS
coor(i5,4)=cpd7(1,i5)-cpd7(2,i5)                         INTERS
coor(i5,6)=cpd7(1,i5)                                    INTERS
end do                                                    INTERS
coor(4,1)=1                                               INTERS
coor(5,2)=1                                               INTERS
coor(4,5)=cpd7(3,1)-cpd7(4,1)                            INTERS
coor(5,5)=cpd7(3,2)-cpd7(4,2)                            INTERS
coor(4,6)=cpd7(3,1)                                     INTERS
coor(5,6)=cpd7(3,2)                                     INTERS
if((cpd7(1,2).eq.cpd7(2,2)).and.(cpd7(3,2).eq.cpd7(4,2)).and.(cpd
+7(1,2).eq.cpd7(3,2))) then                                INTERS
coor(5,2)=0                                               INTERS
coor(5,3)=1                                               INTERS
coor(5,5)=cpd7(3,3)-cpd7(4,3)                            INTERS
coor(5,6)=cpd7(3,3)                                     INTERS
end if                                                    INTERS
if((cpd7(1,1).eq.cpd7(2,1)).and.(cpd7(3,1).eq.cpd7(4,1)).and.(cpd
+7(1,1).eq.cpd7(3,1))) then                                INTERS
coor(4,1)=0                                               INTERS
coor(4,3)=1                                               INTERS
coor(4,5)=cpd7(3,3)-cpd7(4,3)                            INTERS
coor(4,6)=cpd7(3,3)                                     INTERS
end if                                                    INTERS
call gauss(coor,sol,5)                                    INTERS
return                                                    INTERS
end                                                        INTERS
c *****INTERS
c *****MATPA
c           MATRICE DE PASSAGE                            MATPA
c *****MATPA
subroutine matpass(cpd5,p)                                MATPA

```

```

implicit double precision (a-h,o-z)
dimension cpd5(5,3),p(3,4),w1(3),w2(3),w3(3)
do i=1,3
p(i,1)=cpd5(3,i)-cpd5(1,i)
p(i,3)=cpd5(5,i)
end do
xnorme=dsqrt(p(1,1)**2+p(2,1)**2+p(3,1)**2)
do i=1,3
p(i,1)=p(i,1)/xnorme
end do
do i=1,3
w1(i)=p(i,3)
w2(i)=p(i,1)
end do
call prodvec(w1,w2,w3)
do i=1,3
p(i,2)=w3(i)
end do
return
end
c *****MATPA
c *****PRVEEC
e PRODUCT VECTORIEL NORMALISE PRVEEC
c *****PRVEEC
subroutine prodvec(w1,w2,w3)
implicit double precision (a-h,o-z)
dimension w1(3),w2(3),w3(3)
w3(1)=w1(2)*w2(3)-w2(2)*w1(3)
w3(2)=w2(1)*w1(3)-w1(1)*w2(3)
w3(3)=w1(1)*w2(2)-w2(1)*w1(2)
x=dsqrt(w3(1)**2+w3(2)**2+w3(3)**2)
do k=1,3
w3(k)=w3(k)/x
end do
return
end
c *****PRVEEC
c *****PROMV
e SUBROUTINE QUI FAIT LE PRODUIT D'UNE MATRICE AVEC UN VECTEUR PROMV
c *****PROMV

```

```

subroutine      prodmv(a,n,v,prod)
implicit      double      precision(a-h,o-z)
dimension     a(n,n),v(n),prod(n)
do            i=1,n
prod(i)=0.
do            j=1,n
prod(i)=prod(i)+a(i,j)*v(j)
end          do
end          do
return
end
c *****
c *****
c          FUNCTION : FONCTION DE CHRONICKER
c *****
function      chron(ldx,ldy)
implicit      double      precision      (a-h,o-z)
if(ldx.eq.ldy)      then
chron=1.
else
chron=0.
end          if
return
end
c *****
c *****
c          SUBROUTINE :      DISTANCE ENTRE DEUX POINTS X,Y
c *****
subroutine      dis(cpd3,r)
implicit      double      precision      (a-h,o-z)
dimension     cpd3(2,3)
r=0.
do            i=1,3
r=r+(cpd3(1,i)-cpd3(2,i))**2
end          do
r=dsqrt(r)
return
end
c *****

```

```

c *****INVER
c   SUBROUTINE QUI CALCUL L'INVERSE D'UNE MATRICE      INVER
c *****INVER
  subroutine      inverse(p,p1)                          INVER
  implicit double precision (a-h,o-z)                   INVER
  dimension      p(3,4),p1(3,3),v(3),px(3,4)            INVER
  do      j=1,3                                          INVER
  do      i=1,3                                          INVER
  p(i,4)=chron(i,j)                                     INVER
  end                                           do      INVER
  do      i8=1,3                                         INVER
  do      j8=1,4                                         INVER
  px(i8,j8)=p(i8,j8)                                   INVER
  end                                           do      INVER
  end                                           do      INVER
  call      gauss(px,v,3)                               INVER
  do      i=1,3                                          INVER
  p1(i,j)=v(i)                                         INVER
  end                                           do      INVER
  end                                           do      INVER
  return                                             INVER
  end                                             INVER
c *****INVER

c *****GAUSS
c   SUBROUTINE : GAUSS AVEC PIVOTATION TOTALE
c *****GAUSS
c *****GAUSS
c *   ENTREE : l1 : nombre d'equations                  *GAUSS
c *   coor(i,j) i=1,l1 ;j=1,l1 :valeurs de la matrice *GAUSS
c *   coor(i,l1+1) : vecteur B                          *GAUSS
c *   SORTIE : sol(i) : la solution i                   *GAUSS
c *****GAUSS
  subroutine      gauss(coor,sol,l1)                     GAUSS
  implicit double precision (a-h,o-z)                   GAUSS
  dimension      coor(l1,l1+1),sol(l1)                 GAUSS
  dimension      k(5)                                   GAUSS
  do      i=l1                                          GAUSS
  do      15      j=1,l1-1                              GAUSS
  sup=coor(j,j)                                        GAUSS
  l=j                                                  GAUSS

```

```

n=j
do 11 lc=j,11
do 11 l11=j,11
if(abs(sup).ge.abs(coor(l11,lc))) go to 11
sup=coor(l11,lc)
l=l11
n=lc
11 continue
k(j)=n
do 8 kk=j,11+1
t=coor(j,kk)
coor(j,kk)=coor(l,kk)
coor(l,kk)=t
8 continue
do 14 i=1,11
t=coor(i,j)
coor(i,j)=coor(i,n)
coor(i,n)=t
14 continue
do 15 i=j+1,11
do 15 kk=11+1,j,-1
coor(i,kk)=coor(i,kk)-(coor(j,kk)/coor(j,j))*coor(i,j)
15 continue
sol(11)=coor(11,11+1)/coor(11,11)
do 25 i=11-1,1,-1
som=0
do 24 l=11,i+1,-1
som=som+coor(i,l)*sol(l)
24 continue
sol(i)=(coor(i,11+1)-som)/coor(i,i)
25 continue
do 66 i=11-1,1,-1
t=sol(i)
sol(i)=sol(k(i))
sol(k(i))=t
66 continue
return
end
c *****GAUSS GAUSS

```

REFERENCES

- K. J. BATHE : Finite element procedures in engineering analysis.
Prentice-Hall ,I.N.C . 1982
- C. A. BREBBIA : The boundary element method for engineers .
Pentech Press . 1978
- S. L. CROUCH - A. M. STARFIELD : Bondary element method in solid
mechanics.
Allen and Unwin . 1983
- T. A. CRUSE : Numerical solutions in three dimensional
elastostatics .
Int. J. Solids Structures . 1969
- T. A. CRUSE : Three-dimensional elastic stress analysis of a
fracture specimen with an edge crack .
Int. J. of Fracture Mecanics . 1971
- G. DHATT et G. TOUZOT : Une présentation de la méthode des
éléments finis.
Malone S. A. 1984
- P. GERMAIN : Mécanique des milieux continus.
Masson et Cie. 1962
- C. LACHAT : A further development of the boudary integral
technique for elastostatics
Thesis submitted for the degree of DOCTOR OF
PHILOSOPHY . University of SOUTHAMPTON . 1975
- C. LACHAT and O.WATSON :Effective numerical treatment of boundary
integral equations : a formulation for
three-dimensional elastostatics.
Int. J. for Num. Methods in Engineering.
1976

- T. MESSAFER : Coupling of finite element and boundary element techniques.
Thesis submitted for the degree of MASTER OF PHILOSOPHY . University of BIRMINGHAM . 1988
- L. SOLOMON : Elasticité linéaire .
Masson et Cie . 1968
- TIMOSHENKO : Théorie de l'élasticité .
Librairie Polytechnique CH. BERANGER . 1948
- C.E.T.I.M : Application de la methode des equations intégrales a la mécanique.
Publication (1978)