



HAL
open science

Contribution à l'étude du champ de contraintes dans les matériaux composites. Méthode des droites rayonnantes

André Chomarat, J. Savel

► To cite this version:

André Chomarat, J. Savel. Contribution à l'étude du champ de contraintes dans les matériaux composites. Méthode des droites rayonnantes. [Rapport de recherche] 617/89, Ministère de l'équipement et du logement / Bureau de la recherche architecturale (BRA); Ministère de la recherche; Ecole nationale supérieure d'architecture de Nantes. 1989. hal-01906429

HAL Id: hal-01906429

<https://hal.science/hal-01906429v1>

Submitted on 26 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

617

MINISTERE DE L'EQUIPEMENT ET DU LOGEMENT

CONTRIBUTION A L'ETUDE DU CHAMP DE
CONTRAINTES DANS LES MATERIAUX COMPOSITES

METHODE DES DROITES RAYONNANTES

A.Chomarat J.Savel

Ecole d'architecture de NANTES

Novembre 1989

Ministère de l'Équipement et du Logement
Direction de l'Architecture et de l'Urbanisme
Sous Direction des Enseignement et des Professions
Bureau de la Recherche Architecturale.

Le présent document constitue le rapport final d'une recherche remise au Bureau de la Recherche Architecturale en exécution du programme général de recherche mené par le Ministère de l'Équipement et du Logement avec le Ministère de la Recherche.
Les jugements et opinions émis par les responsables de la recherche n'engagent que leurs auteurs.

RAPPORT DE RECHERCHE

Contrat n° 88.01234.00.223.75.01

Contribution à l'étude du champ de
contraintes dans les matériaux composites

Méthode des droites rayonnantes

A.Chomarat

J.savel

SOMMAIRE

1. Principe de la méthode et hypothèses de Calculs
2. Principe de Calcul
 - 2.1 Recherche des déplacements en un point quelconque du solide
 - 2.2 Détermination des contraintes en M
3. Formulation de la méthode
 - 3.1 Données géométriques du contour
 - 3.2 Résolution de la segmentation
 - 3.2.1 Coordonnées du point d'intersection
 - 3.2.2 Cas particulier
 - 3.2.3 Solution des points d'intersection
 - 3.3 Détermination des deux points les plus proches de M
 - 3.4 Calcul des déplacements du point M
 - 3.5 Exploration du domaine
 - 3.6 Relations de la théorie de l'élasticité linéaire
 - 3.7 Relations contraintes - déformations en élasticité linéaire
4. Informatisation de la Méthode. Description Technique du programme PLQ
 - 4.1 Programme PLQ
 - 4.2 Développements informatiques futurs
5. Mode d'emploi du logiciel PLQ
 - 5.1 Entrées
 - 5.2 Sorties
6. Validité de la méthode. Cas classiques de l'élasticité linéaire
7. Etude comparative avec la méthode des éléments finis

1 PRINCIPE DE LA METHODE

1. Principe de la méthode et hypothèses de calculs.

Les matériaux étudiés sont continus, homogènes, isotropes et travaillent dans le domaine élastique. Lorsque ces conditions sont requises, deux hypothèses théoriques ont été émises et servent de base à la méthode.

Hypothèse 1.

Dans tout matériau, le déplacement, dans une direction quelconque, d'un point M d'un segment de droite, appartenant en totalité au même matériau est obtenu par combinaison linéaire des déplacements, dans la même direction, de deux autres points I et J du segment.

Hypothèse 2.

Dans tout matériau, le déplacement résultant d'un point M dans une direction quelconque est égal à la moyenne des déplacements dans cette même direction, de tous les points de même coordonnées que M, appartenant aux segments de droite passant par ce point.

Ces deux hypothèses sont appliquées au cas d'un solide composé d'un seul matériau ou de plusieurs matériaux de caractéristiques mécaniques différentes (composites).

Pour chacun des cas il est admis que les déplacements sont connus, sur le contour du solide ou sur les interfaces des composants dans le cas particulier d'un composite.

En un premier temps l'étude a été limitée aux cas des solides plans à contours polygonaux sollicités suivant un état de contraintes planes ou un état de déformations planes.

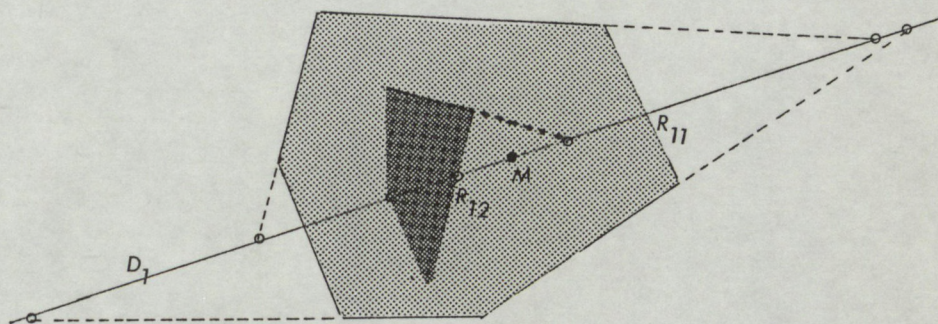
La généralisation dans l'espace ne présente pas de difficultés théoriques particulières mais sa mise en équation est évidemment beaucoup plus laborieuse.

2 PRINCIPE DE CALCUL

2. Principe de calcul.

2.1 Recherche des déplacements en un point M quelconque du solide.

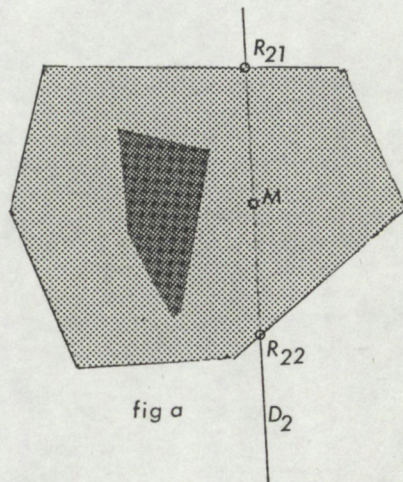
Une droite D_1 passant par un point M situé à l'intérieur du solide coupe le contour en deux points R_{11} et R_{12} situés de part et d'autre de M et tels que le segment $R_{11} R_{12}$ est entièrement situé dans un même matériau.



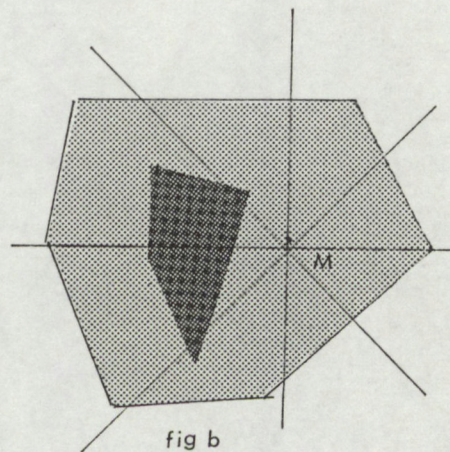
Le contour étant polygonal cette même droite va couper tous les autres cotés ou leurs prolongements (sauf dans le cas particulier où elle est parallèle à l'un des cotés) en d'autres points qu'il convient d'éliminer dans une opération de sélection. En effet le critère retenu dans les hypothèses de départ implique que le segment $R_{11} M R_{12}$ soit contenu dans un même matériau et que les points d'intersections soient situés sur le contour. Pour tous les points d'intersections répondant à cette dernière condition, le critère de continuité du matériau imposé au segment $R_{11} M R_{12}$ implique que les valeurs absolues des distances MR_{11} et MR_{12} soient les deux valeurs minima parmi toutes les distances des segments séparant M des différents points d'intersection.

Les déplacements des points du contour donc de R_{11} et R_{12} , étant connus par hypothèse, la détermination des déplacements du point M appartenant au segment $R_{11} R_{12}$ peut s'effectuer à partir de l'hypothèse 1 précédemment formulée.

Une seconde droite D_2 passant par le même point M permet par un processus analogue de sélectionner les points R_{21} et R_{22} et d'en déduire les déplacements du point M . (fig. a)



L'opération est répétée n fois, de façon telle que les droites passant par le point M fassent entre elles des angles égaux. (fig. b)



La valeur du paramètre définissant le nombre de droites est déterminée en fonction de la précision de calcul retenue et de la complexité du contour (ce paramètre est introduit dans la première ligne des entrées sous le nom n ray).

Le déplacement résultant du point M est obtenu, à partir des résultats précédents par application de l'hypothèse 2.

2.2 Détermination des contraintes en M.

Sans que cela soit une obligation mais pour des raisons pratiques de programmation l'exploration du domaine est effectuée sur un maillage de pas p et q dans deux directions perpendiculaires (celle des axes généraux). (fig. c)

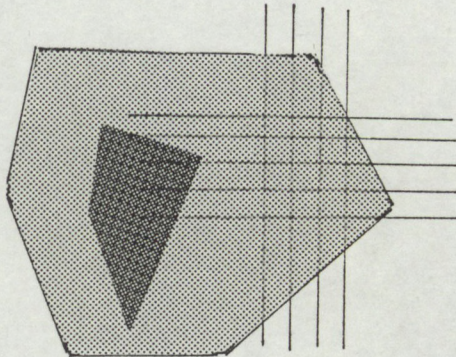


fig c

La connaissance des déplacements en chaque point du maillage permet de déterminer les variations de ces déplacements dans deux directions perpendiculaires et d'en déduire ainsi, grâce aux relations élémentaires de l'élasticité linéaire, les déformations unitaires longitudinales et angulaires en chaque point du maillage.

Les relations contraintes déformations établies suivant les cas, en déformations planes ou en contraintes planes permettent d'accéder aux valeurs des contraintes normales et tangentielles en chaque point du maillage.

3 FORMULATION DE LA METHODE

3. Formulation de la méthode.

3.1 Données géométriques du contour.

ns : nombre de sommets

nb : nombre de cotés

coordonnées des sommets : $S_i (x_i, y_i)$ pour $1 \leq i \leq ns$

cotes du contour. (sommets liés) $S(i_k, j_k)$ pour $1 \leq k \leq nb$

Déplacements des points du contour

$u_k = f_k(m)$ pour $0 \leq m \leq 1$

$v_k = g_k(m)$ pour $1 \leq k \leq nb$

E module d'élasticité longitudinal

V coefficient de Poisson

3.2 Résolution de segmentation.

Point M de coordonnées (x_m, y_m)

Droite de pente α passant par le point M

3.2.1 Coordonnées du point d'intersection de la droite avec un coté k du contour.

$$x_k = (a_k x_{ik} - x_m \operatorname{tg} \alpha + y_m - y_{ik}) / (a_k - \operatorname{tg} \alpha)$$

$$y_k = (a_k (x_{ik} - x_m) \operatorname{tg} \alpha + a y_m - y_{ik} \operatorname{tg} \alpha) / (a_k - \operatorname{tg} \alpha)$$

$$a_k = (y_{jk} - y_{ik}) / (x_{jk} - x_{ik})$$

$$0 \leq k \leq nb$$

3.2.2 Cas particulier.

$a_k = \operatorname{tg} \alpha$ droite parallèle à l'un des côtés, pas de point d'intersection

$$\alpha = \pi/2$$

$$x_k = x_m$$

$$y_k = a_k x_m - a_k x_i = y_i$$

3.2.3 Sélection des points d'intersections.

Les points sont éliminés lorsque l'une des conditions ci-dessous est vérifiée

$x_{jk} > x_{ik}$ si $x_k > x_{jk}$ ou si $x_k < x_{ik}$
 $x_{jk} = x_{ik}$ si $y_{jk} > y_{ik}$ et si $y_k > y_{jk}$ ou si $y_k < y_{ik}$
si $y_{jk} < y_{ik}$ et si $y_k > y_{ik}$ ou si $y_k < y_{jk}$
 $x_{jk} < x_{ik}$ si $x_k > x_{ik}$ ou si $x_k < x_{jk}$

En fin de sélection on obtient p points de coordonnées :

$$(x_k, y_k)_p$$

Elimination des points doubles.

si $x_{kp} = x_{k'p}$ et si $y_{kp} = y_{k'p}$ le point p' n'est pas retenu.

Les égalités étant à une petite valeur près ϵ .

On arrive à R points sélectionnés de coordonnées:

$$(x_{kr}, y_{kr}) \text{ avec } r = 2, 3, 4, \dots$$

3.3 Détermination des deux points les plus proches du point initial M.

Coordonnées de M : (x_M, y_M)

Coordonnées de R : (x_{kr}, y_{kr}) avec $r = 2, 3, 4, \dots$

longueur du segment MR

$$dMR = [(x_{kr} - x_M)^2 + (y_{kr} - y_M)^2]^{1/2}$$

$\alpha \neq \pi/2$

$x_{kr} > x_M$ r_1 nombre de points répondant à cette condition

calculer $(dMR)_i$ avec $1 < i < r_1$
garder la plus petite valeur.

$x_{kr} < x_M$ r_2 nombre de points répondant à cette condition

calculer $(dMR)_i$ avec $1 < i < r_2$
garder la plus petite valeur

$\alpha = \pi/2$

$y_{kr} > y_M$ r_1 nombre de points répondant à cette condition

calculer $(dMR)_i$ avec $1 < i < r_1$
garder la plus petite valeur

$y_{kr} < y_M$ r_2 nombre de points répondant à cette condition

calculer $(dMR)_i$ avec $1 < i < r_2$
garder la plus petite valeur

En fin de sélection il reste deux points.

3.4 Calcul des déplacements du point M.

On désigne par:

x_{k1}, y_{k1} et x_{k2}, y_{k2} les coordonnées des deux points R_{k1} et R_{k2} retenus

$(dMR)_1$ et $(dMR)_2$ les distances des segments MR_{k1} et MR_{k2} précédemment calculés.

U_M, V_M les déplacements du point M appartenant au segment $R_{k1} R_{k2}$

$$U_M = [(dMR)_1 / ((dMR)_1 + (dMR)_2)] U_{k2} + [(dMR)_2 / ((dMR)_1 + (dMR)_2)] U_{k1}$$

$$V_M = [(dMR)_1 / ((dMR)_1 + (dMR)_2)] V_{k2} + [(dMR)_2 / ((dMR)_1 + (dMR)_2)] V_{k1}$$

Par le point M l'opération est répétée n fois en donnant à l'angle α les valeurs:

$$\alpha = 0, \pi/n, 2\pi/n, 3\pi/n \dots (n-1)\pi/n$$

Déplacements résultants

$$\bar{U}_M = 1/n \sum_0^n U_M$$

$$\bar{V}_M = 1/n \sum_0^n V_M$$

3.5 Exploration du domaine

p et q représentant les pas suivant ox et oy

pour M (x_M, y_M)

les déplacements ont pour valeurs \bar{U}_M, \bar{V}_M

pour M (x_{M-p}, y_{M-p})

les déplacements ont pour valeurs $\bar{U}_{M-p}, \bar{V}_{M-p}$

pour M (x_{M+p}, y_{M-p})

les déplacements ont pour valeurs \bar{U}_{M+P} , \bar{V}_{M+P}
d'où les valeurs, des dérivées partielles, que l'on désigne
par $\delta\bar{U}_M / \delta x$ et $\delta\bar{V}_M / \delta x$ obtenues par interpolation
parabolique (formule de Lagrange).

De même dans la direction Oy on obtient :

$$\delta\bar{U}_M / \delta y \text{ et } \delta\bar{V}_M / \delta y$$

3.6 Relations de la théorie de l'élasticité linéaire.

e_x déformation longitudinale dans la direction Ox

e_y déformation longitudinale dans la direction Oy

δ_{xy} déformation angulaire

$$e_x = \delta\bar{U}_M / \delta x \quad e_y = \delta\bar{V}_M / \delta y \quad \delta_{xy} = (\delta\bar{U}_M / \delta y + \delta\bar{V}_M / \delta x) / 2$$

3.7 Relations contraintes - déformation en élasticité linéaire

Etat de contraintes planes $n_z = 0$

$$n_x = E (e_x + \nu e_y) / (1 - \nu^2)$$

$$n_y = E (e_y + \nu e_x) / (1 - \nu^2)$$

$$t_{xy} = E \delta_{xy} / (2(1 + \nu))$$

Etat de déformations planes $e_z = 0$

$$n_x = E ((1 - \nu)e_x + \nu e_y) / ((1 + \nu)(1 - 2\nu))$$

$$n_y = E ((1 - \nu)e_y + \nu e_x) / ((1 + \nu)(1 - 2\nu))$$

$$t_{xy} = E \delta_{xy} / (2(1 + \nu))$$

avec

n_x - contrainte normale s'exerçant sur une facette perpendiculaire à Ox.

n_y - contrainte normale s'exerçant sur une facette perpendiculaire à Oy.

t_{xy} - contrainte tangentielle s'exerçant sur une facette perpendiculaire à Ox.

4 INFORMATISATION DE LA METHODE

Page 10
manquante sur le rapport

4 Informatisation de la méthode . Description technique du programme PLQ.

outils utilisés sont ceux habituellement utilisés dans les programmes basés sur des concepts géométriques. (Intersection de droites, problèmes de proximité.....).

4.1 Programme

Le programme est entièrement écrit en langage C et peut être utilisé sous les système d'exploitation MS/DOS ou UNIX.

Ce programme suit strictement la méthode théorique exposée dans le paragraphe 3.

Un certain nombre de points sont à noter: la représentation des résultats.

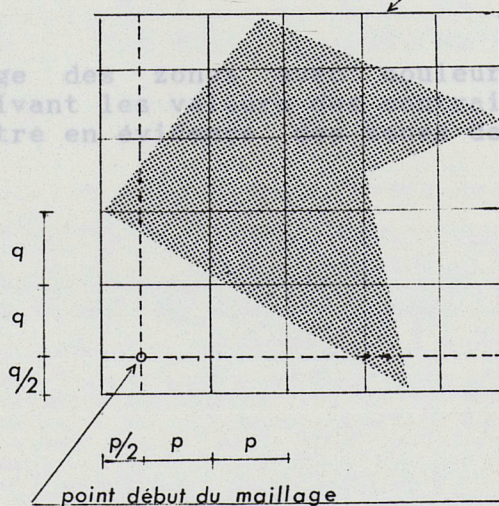
- Les calculs sont effectués aux sommets d'un maillage dont les bornes sont définies par le contour polygonal de la plaque. Les pas de ce maillage ont respectivement pour valeurs p dans la direction des x et q dans la direction des y .

On recherche dans un premier temps le plus petit rectangle de côtés parallèles aux axes et enveloppant la plaque. L'origine du maillage étant au point bas gauche de ce rectangle décalé d'un demi pas respectivement en x et en y .

- Les coefficients calculés des droites, supports des côtés du polygone figurant le contour sont stockés en mémoire pour être utilisés à différents moments en cours de calcul.

- En chaque point du maillage précédemment défini, les dérivées des déplacements sont déterminés par interpolation parabolique à l'aide des formules de Lagrange.

rectangle enveloppe



- repérage de zones ou hachurages différents suivant les contraintes. Ce système permet de mettre en évidence les majorations des contraintes.

La structure du programme n'appelle pas de remarques spéciales. Les outils utilisés sont ceux habituellement utilisés dans les programmes basés sur des concepts géométriques. (Intersection de droites, problèmes de proximité.....).

La méthode utilisée pour déterminer si un point est à l'intérieur ou non d'un polygone est celle dite "du nombre d'intersections".

4.2 Développements informatiques futurs.

Des améliorations du programme, sont surtout à envisager au niveau de la saisie des données et de la représentation des résultats.

- Saisie des données à partir d'un éditeur graphique basé sur un digitaliseur avec témoin sur écran. Ce système doit s'avérer particulièrement intéressant pour la saisie des déplacements qui peut ainsi s'effectuer directement en utilisant un coefficient amplificateur.

- Représentation graphique des résultats.

Dans le programme actuel, les contraintes, longitudinales, tangentiels, principales et l'orientation de ces dernières sont données sous forme de tableaux. Dans une version plus pédagogique il est nécessaire d'envisager une représentation graphique des résultats sur traceur selon plusieurs formes au choix de l'utilisateur. Selon les cas de sollicitations on peut en effet tracer:

- des vecteurs symbolisant des contraintes principales (norme du vecteur proportionnelle à la grandeur de la contrainte) ce mode de représentation présente l'avantage de symboliser les lignes isostatiques.

- repérage des zones avec couleurs ou hachurages différents suivant les valeurs des contraintes. Ce système permet de mettre en évidence des zones de majorations des contraintes.

5 MODE D'EMPLOI DU LOGICIEL

5. Mode d'emploi du logiciel P.L.Q.

- A l'aide d'un éditeur de texte, constituer un fichier texte dont les données de chaque ligne sont séparées au moins par un espace.
- Les unités de mesure des paramètres doivent être cohérentes.

5.1 Entrées

- Les paramètres d'entrée sont les suivants:

Première ligne	p	q	nray	v	E	
	p	pas dans la direction des x				
	q	pas dans la direction des y				
	nray	nombre de droites de calcul, passant par chaque point				
	v	coefficient de Poisson				
	E	module d'élasticité				

Remarque: le nombre de droites passant par M dépend de la précision recherchée et de la zone étudiée dans le cas des matériaux composites (voir paragraphe 2.1). Dans les exemples traités nous avons en règle générale choisi $nray = 8$. Au delà le temps de calcul augmente sans apporter, sauf cas exceptionnel, des résultats plus intéressants.

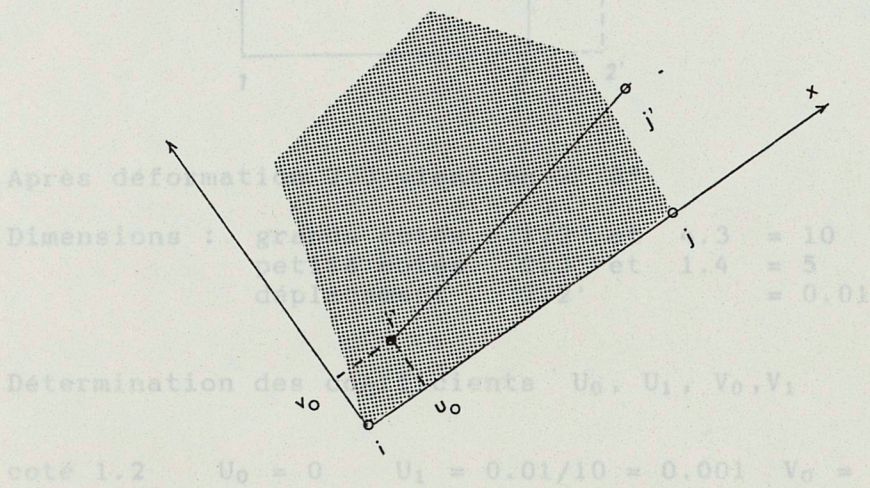
Deuxième ligne	nbs	nombre de sommets du contour				
Lignes suivantes	x	y	coordonnées des sommets			
	"	"	"			
	"	"	"			
	"	"	"			
Ligne suivante	nb cotés	nombre de cotés du contour				
Lignes suivantes	début	fin	u_0	v_0	u_1	v_1
	"	"	"	"	"	"

où début et fin sont les numéros des sommets extrémités de chaque cotés du contour polygonal.

u_0 , v_0 , u_1 , v_1 définissent les déplacements des extrémités des cotés du contour comme il est indiqué ci-après.

A chaque coté ij du contour est associé un repère orthonormé X, Y tel que l'origine soit fixé au point début du coté (défini dans les entrées précédentes) et l'axe des X dirigé dans le sens début-fin.

L'axe Y perpendiculaire au précédent est orienté dans le sens direct.



Après déformation le coté initialement placé en ij vient en $i'j'$.

Dans la version actuelle on admet par hypothèse que les déplacements de chaque point du coté peuvent s'exprimer sous forme linéaire selon les formules

$$U = U_0 + U_1 X$$

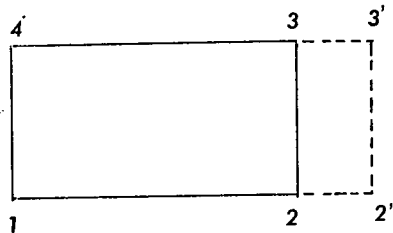
$$V = V_0 + V_1 Y$$

U déplacement d'un point d'un coté suivant la direction X
 V déplacement d'un point d'un coté suivant la direction Y
 U_0 V_0 représentent les déplacements du point début i dans le repère local XY

U_1 V_1 les paramètres déterminés à partir des déplacements imposés au point j

Exemple : plaque en traction

contour initial 1.2.3.4



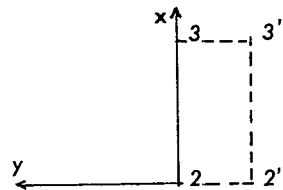
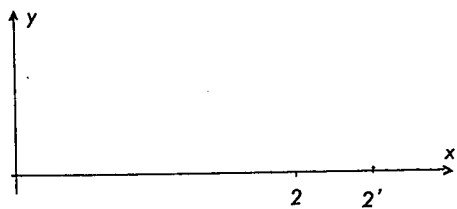
Après déformation 2.3 vient en 2' 3'

Dimensions : grands cotés 1.2 et 4.3 = 10
 petits cotés 2.3 et 1.4 = 5
 déplacement 2.2' = 0.01

Détermination des coefficients U_0, U_1, V_0, V_1

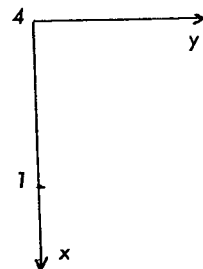
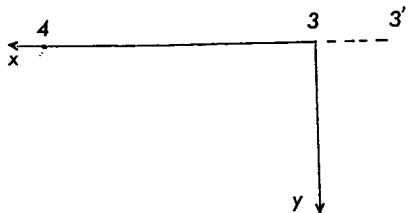
coté 1.2 $U_0 = 0$ $U_1 = 0.01/10 = 0.001$ $V_0 = 0$ $V_1 = 0$

coté 2.3 $U_0 = 0$ $U_1 = 0$ $V_0 = -0.01$ $V_1 = 0$



coté 3.4 $U_0 = -0.01$ $U_1 = 0.01:10 = 0.01$ $V_0 = 0$ $V_1 = 0$

coté 4.1 $U_0 = 0$ $U_1 = 0$ $V_0 = 0$ $V_1 = 0$



Récapitulatif des entrées.

```
p   q   nray   v   E
nbs
x   y
"   "
"   "
nb cotés

début   fin   U0   V0   U1   V1
"       "       "       "       "       "
"       "       "       "       "       "
"       "       "       "       "       "
```

Ces entrées sont stockées dans un fichier dont le nom est demandé en début d'exécution du programme.

L'utilisateur devra également préciser:

- Le mode de calcul retenu. Etat de contraintes planes ou Etat de déformations planes .

- La fenêtre de sortie des points de calcul (fig a). Par défaut le logiciel propose la fenêtre qui englobe exactement le contour (fig b).

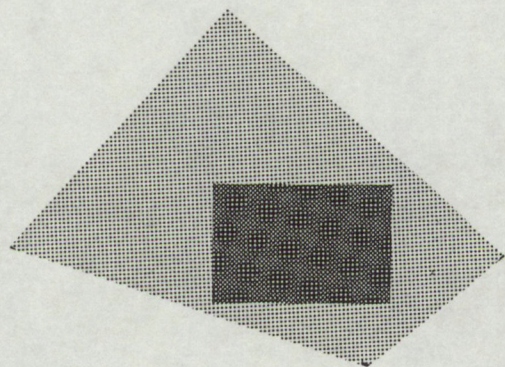


fig a

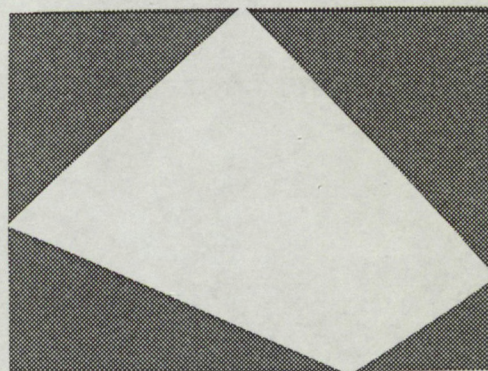


fig b

Cette notion de fenêtre s'avère particulièrement intéressante lorsque l'utilisateur souhaite préciser certaines valeurs de contraintes en un point précis du solide.

Une fenêtre est définie par les coordonnées de son sommet bas gauche et de son sommet droit (fig c).

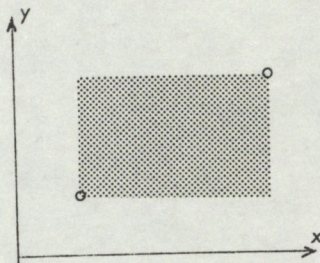


fig c

5.2 Sorties

En sortie sur l'imprimante le programme affiche en premier lieu les données de l'étude puis sous forme d'un tableau

X et Y	les coordonnées du point de calcul considéré
U et V	les déplacements par rapport au repère général
n_x n_y t_{xy}	contraintes normales s'exerçant sur les facettes respectivement perpendiculaires à Ox et Oy et la contrainte tangentielle.
n_1 , n_2 , β	contraintes principales au point considéré et angle de la contrainte principale n_1 avec l'axe Ox . (Cette sortie n'existe que dans la dernière version du logiciel et n'apparaîtra pas, dans tous les résultats que nous donnerons ultérieurement).

6 VALIDITE DE LA METHODE

6 Validité de la méthode. Cas classiques de l'élasticité linéaire.

Les premiers contrôles effectués ont permis de vérifier l'exactitude de la transcription des propositions théoriques formulées dans le paragraphe 3. En particulier pour les contours simplement connexes où la sélection des points d'intersections nécessite une programmation très affinée.

Dans un premier temps il convient de vérifier la validité des hypothèses servant de base aux calculs. Ces vérifications portent sur les cas classiques de la résistance des matériaux pour lesquels les résultats sont parfaitement connus.

Nom du fichier : pl

p = 1.000 q = 1.000 nray = 8 nu = 0.30 E = 20000.0

nombre de sommets : 4

coordonnées des sommets :

Numéro	X	Y
0	0.0000	0.0000
1	4.0000	0.0000
2	4.0000	4.0000
3	0.0000	4.0000

nombre de cotés : 4

Conditions imposées sur le contour

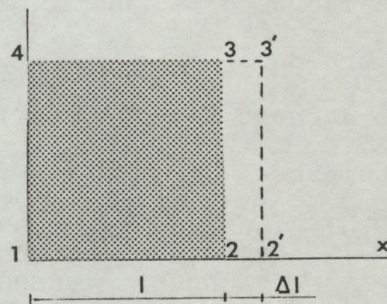
début	fin	u0	u1	v0	v1
0	1	0.0000	0.0250	0.0000	0.0000
1	2	0.0000	0.0000	-0.1000	0.0000
2	3	-0.1000	0.0250	0.0000	0.0000
3	0	0.0000	0.0000	0.0000	0.0000

RESULTATS

Calculs en Contraintes planes

X	Y	u	v	nx	ny	txy	n1	n2	beta
0.50	0.50	0.01250	0.00000	549.45	164.84	-0.00	549.45	164.84	-0
1.50	0.50	0.03750	0.00000	549.45	164.84	-0.00	549.45	164.84	-0
2.50	0.50	0.06250	0.00000	549.45	164.84	-0.00	549.45	164.84	-0
3.50	0.50	0.08750	0.00000	549.45	164.84	0.00	549.45	164.84	0
0.50	1.50	0.01250	0.00000	549.45	164.84	0.00	549.45	164.84	0
1.50	1.50	0.03750	0.00000	549.45	164.84	0.00	549.45	164.84	0
2.50	1.50	0.06250	0.00000	549.45	164.84	0.00	549.45	164.84	0
3.50	1.50	0.08750	0.00000	549.45	164.84	0.00	549.45	164.84	0
0.50	2.50	0.01250	0.00000	549.45	164.84	0.00	549.45	164.84	0
1.50	2.50	0.03750	0.00000	549.45	164.84	0.00	549.45	164.84	0
2.50	2.50	0.06250	0.00000	549.45	164.84	0.00	549.45	164.84	0
3.50	2.50	0.08750	0.00000	549.45	164.84	0.00	549.45	164.84	0
0.50	3.50	0.01250	0.00000	549.45	164.84	0.00	549.45	164.84	0
1.50	3.50	0.03750	0.00000	549.45	164.84	0.00	549.45	164.84	0
2.50	3.50	0.06250	0.00000	549.45	164.84	0.00	549.45	164.84	0
3.50	3.50	0.08750	0.00000	549.45	164.84	0.00	549.45	164.84	0

6.1 Plaque carrée soumise à une traction dans la direction Ox. fichier P1.
cotés parallèles à Ox bloqués



6.1.1 Valeurs théoriques.

$$e_x = \Delta l / l \quad e_y = 0 \quad \text{le coté numéroté 4 3 est bloqué}$$

Etat de contraintes planes

$$n_x = E (e_x + \nu e_y) / (1 - \nu^2) = E e_x / (1 - \nu^2)$$

$$n_y = E (e_y + \nu e_x) / (1 - \nu^2) = \nu E e_x / (1 - \nu^2)$$

$$t_{xy} = 0$$

Etat de déformations planes

$$n_x = E (1 - \nu) e_x / ((1 + \nu)(1 - 2\nu))$$

$$n_y = E \nu e_x / ((1 + \nu)(1 - 2\nu))$$

Application numérique

$$l = 4 \quad \Delta l = 0,1 \quad \nu = 0,3 \quad E = 20.000 \text{ daN/mm}^2$$

Etat de contraintes planes

$$n_x = 549,45 \text{ daN/mm}^2$$

$$n_y = 164,84 \text{ daN/mm}^2$$

$$t_{xy} = 0$$

Etat de déformations planes

$$n_x = 673,08 \text{ daN/mm}^2$$

$$n_y = 288,46 \text{ daN/mm}^2$$

$$t_{xy} = 0$$

6.1.2 Valeurs obtenues avec le programme PLQ.

Valeurs identiques. voir tableau ci-contre.

Nom du fichier : p2

p = 1.000 q = 1.000 nray = 8 nu = 0.30 E = 20000.0

nombre de sommets : 4

coordonnées des sommets :

Numéro	X	Y
0	0.0000	0.0000
1	4.0000	0.0000
2	4.0000	4.0000
3	0.0000	4.0000

nombre de cotés : 4

Conditions imposées sur le contour

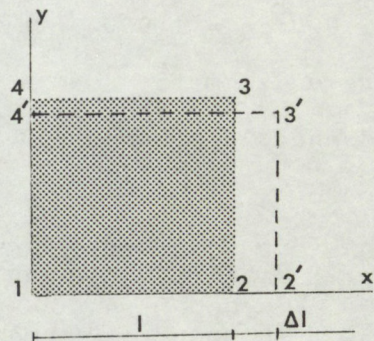
début	fin	u0	u1	v0	v1
0	1	0.0000	0.0250	0.0000	0.0000
1	2	0.0000	-0.0075	-0.1000	0.0000
2	3	-0.1000	0.0250	0.0300	0.0000
3	0	0.0300	-0.0075	0.0000	0.0000

RESULTATS

Calculs en Contraintes planes

X	Y	u	v	nx	ny	txy	n1	n2	beta
0.50	0.50	0.01250	-0.00375	500.00	-0.00	-0.00	500.00	-0.00	-0
1.50	0.50	0.03750	-0.00375	500.00	0.00	-0.00	500.00	0.00	-0
2.50	0.50	0.06250	-0.00375	500.00	0.00	0.00	500.00	0.00	0
3.50	0.50	0.08750	-0.00375	500.00	0.00	-0.00	500.00	0.00	-0
0.50	1.50	0.01250	-0.01125	500.00	-0.00	0.00	500.00	-0.00	0
1.50	1.50	0.03750	-0.01125	500.00	0.00	0.00	500.00	0.00	0
2.50	1.50	0.06250	-0.01125	500.00	0.00	0.00	500.00	0.00	0
3.50	1.50	0.08750	-0.01125	500.00	-0.00	-0.00	500.00	-0.00	-0
0.50	2.50	0.01250	-0.01875	500.00	0.00	0.00	500.00	0.00	0
1.50	2.50	0.03750	-0.01875	500.00	0.00	0.00	500.00	0.00	0
2.50	2.50	0.06250	-0.01875	500.00	0.00	0.00	500.00	0.00	0
3.50	2.50	0.08750	-0.01875	500.00	-0.00	0.00	500.00	-0.00	0
0.50	3.50	0.01250	-0.02625	500.00	0.00	0.00	500.00	0.00	0
1.50	3.50	0.03750	-0.02625	500.00	0.00	0.00	500.00	0.00	0
2.50	3.50	0.06250	-0.02625	500.00	0.00	0.00	500.00	0.00	0
3.50	3.50	0.08750	-0.02625	500.00	-0.00	0.00	500.00	-0.00	0

6.2 Plaque carrée soumise à une traction simple dans la direction Ox fichier P2 cotés parallèles à Oy libre.



6.2.1 Valeurs théoriques

$$e_x = \Delta l / l \quad e_y = -v \Delta l / l$$

avec longueur initiale
L longueur après déformation

Etat de contraintes planes

$$\sigma_x = E (e_x + v e_y) / (1 - v^2) = E e_x$$

$$\sigma_y = E (e_y + v e_x) / (1 - v^2) = 0$$

$$\tau_{xy} = 0$$

Etat de déformations planes

$$\epsilon_x = E e_x (1 - v - v^2) / (1 - v - 2v^2)$$

$$\epsilon_y = E e_x v^2 / (1 - v - 2v^2)$$

Applications numériques

$$l = 4 \quad l = 0,1 \quad v = 0,3 \quad E = 20.000 \text{ daN/mm}^2$$

Etat de contraintes planes

$$\sigma_x = 500 \text{ daN/mm}^2$$

$$\sigma_x = 586,53 \text{ daN/mm}^2$$

$$\sigma_y = 0$$

$$\sigma_y = 86,53 \text{ daN/mm}^2$$

$$\tau_{xy} = 0$$

$$\tau_{xy} = 0$$

6.2.2 Valeurs obtenus avec le programme PLQ.

Valeurs identiques voir tableau ci-contre

Nom du fichier : p3

p = 1.000 q = 1.000 nray = 8 nu = 0.30 E = 20000.0

nombre de sommets : 4

coordonnées des sommets :

Numéro	X	Y
0	0.0000	0.0000
1	4.0000	0.0000
2	4.0000	4.0000
3	0.0000	4.0000

nombre de cotés : 4

Conditions imposées sur le contour

début	fin	u0	u1	v0	v1
0	1	0.0000	0.0000	0.0000	0.0250
1	2	0.1000	0.0000	0.0000	0.0000
2	3	0.0000	0.0000	-0.1000	0.0250
3	0	0.0000	0.0000	0.0000	0.0000

RESULTATS

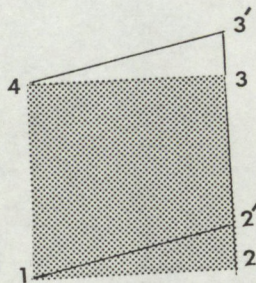
Calculs en Contraintes planes

X	Y	u	v	nx	ny	txy	n1	n2	beta
0.50	0.50	0.00000	0.01250	-0.00	-0.00	192.31	192.31	-192.31	45
1.50	0.50	0.00000	0.03750	-0.00	-0.00	192.31	192.31	-192.31	45
2.50	0.50	0.00000	0.06250	-0.00	-0.00	192.31	192.31	-192.31	45
3.50	0.50	0.00000	0.08750	0.00	0.00	192.31	192.31	-192.31	45

Page 21
manquante sur le rapport

6.3 Plaque soumise à un cisaillement pur.

6.3.1 Valeurs théoriques.



$$\tau_{xy} = E \beta_{xy} / (2(1+\nu))$$

$$\sigma_x = 0$$

$$\sigma_y = 0$$

Avec β_{xy} Angle correspondant à la variation de l'angle droit.

Application numérique

$$\beta_{xy} = 0.1/4$$

$$\tau_{xy} = 192,31 \text{ daN/mm}^2$$

6.3.2 Valeurs obtenues avec le programme PLQ.

Valeurs identiques voir tableau ci-contre

6.4 Conclusion.

Pour les solides constitués d'un seul matériau, les résultats théoriques et ceux obtenus expérimentalement sont rigoureusement identiques. Ceci confirme que l'hypothèse 1, servant de base à la méthode est implicitement contenue dans la théorie de l'élasticité linéaire. Elle pourrait donc être appliquée à tous les cas de figures; cependant la difficulté essentielle est la détermination des déplacements des points du contour. Ces déplacements bien connus dans les cas classiques que nous venons d'aborder, le sont moins lorsque le contour est plus complexe.

7 ETUDE COMPARATIVE

7 Etude comparative - Méthode des éléments finis.

Les résultats théoriques concernant les matériaux composites, n'ayant pas un caractère de référence comme ceux de la théorie de l'élasticité rappelés dans les paragraphes précédents, une étude comparative a été effectuée en utilisant sur le même modèle, une méthode basée sur la théorie des éléments finis.

Modèle retenu :

Le modèle composite étudié est constitué d'un matériau homogène (matrice) de module d'élasticité longitudinal E_m de forme carrée ABCD dans lequel est inclus en son centre un matériau (fibre) de module E_f dont la forme est également un carré a,b,c,d.

Dans un premier essai, on admet par hypothèse, que le rapport E_f/E_m est infiniment grand. Ainsi les déplacements relatifs des points a,b,c,d sont nuls lorsque la matrice est soumise à un effet de traction parallèle à l'un des côtés du carré. Figure a

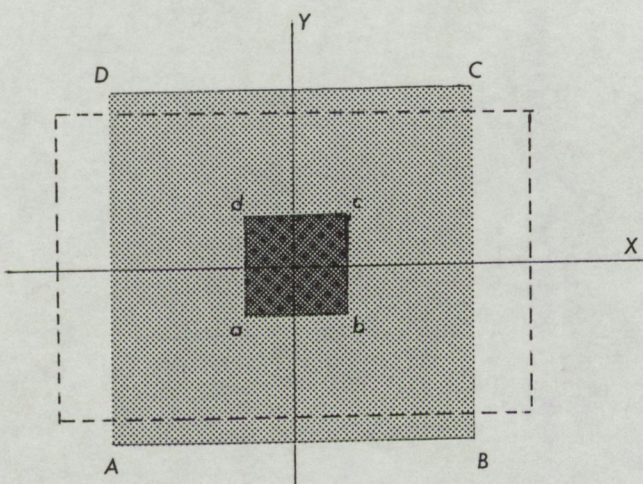


Figure a.

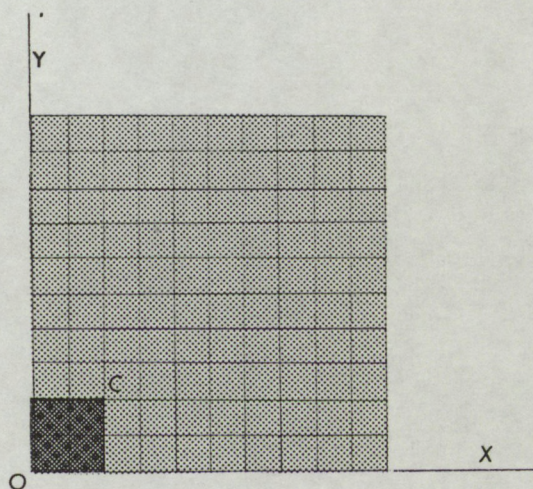


Figure b.

Dans notre méthode, l'effort de traction est traduit par des translations respectivement parallèle à Ox et Oy pour les côtés du carré.

L'étude comparative a été effectuée au laboratoire de Mécanique des Structures de l'Ecole Nationale Supérieure de Mécanique de Nantes, à l'aide du logiciel SAMCEF.

En raison de la symétrie du modèle un quart de la plaque a été étudiée, le maillage est effectué en prenant 100 éléments carrés dont 4 pour le noyau central. Figure b.

La méthode des éléments finis permet d'obtenir les contraintes au centre de chaque élément. Dans les zones pour lesquelles le gradient des contraintes est important il est évident qu'il conviendrait, pour une étude plus approfondie, d'affiner le maillage.

Avec la deuxième méthode, l'étude a été effectuée en prenant un pas égal à la dimension des éléments finis, ce qui permet d'obtenir les contraintes aux mêmes points en raison du décalage d'un demi-pas pris en compte dans la détermination de l'origine du maillage.

Résultats comparatifs.

Pour le modèle étudié les résultats des deux méthodes sont dans l'ensemble assez proches les uns des autres, en particulier, et cela est évident dans les zones les plus éloignées du noyau central.

Résultats généraux.

Sur les tableaux suivant sont portés les résultats des deux méthodes concernant les contraintes normales n_x et n_y ainsi que les contraintes tangentielles n_{xy} .

Les valeurs des contraintes nominales sans noyau central sont

$$n_x = 200 \quad n_y = 0 \quad n_{xy} = 0$$

Contrainte normale nx

204	203	201	199	198	197	197	198	198	198
203	202	200	199	198	198	198	198	199	199
202	200	199	197	197	197	198	199	200	201
198	197	196	195	196	198	200	203	204	205
191	191	191	192	196	201	204	207	209	209
177	179	185	190	200	207	211	213	215	215
136	165	175	200	212	218	220	221	222	222
5,6	7	241	236	234	233	231	230	229	229
		298	267	254	246	241	237	235	234
		263	272	264	254	247	241	238	237

Méthode des éléments finis

199	199	198	197	199	200	199	198	197	203
196	200	198	198	199	199	198	194	200	207
200	192	204	195	190	199	197	199	207	206
195	200	201	180	189	197	204	207	207	205
187	208	180	180	199	213	210	198	204	205
189	179	180	191	212	207	198	209	212	206
189	179	180	191	212	207	198	209	212	206
136	197	185	195	209	203	221	219	207	207
41	158	235	228	233	226	222	207	206	206
		489	271	236	231	214	214	213	213
		381	324	245	234	228	218	215	206

Méthode proposée

Contrainte normale ny

-18	-16	-12	-6,5	-1	3,2	6,6	9	10	11
-20	-17	-12	-6	-5,8	3,8	7	9	10	11
-22	-18	-12	-5	0	5	7,7	9,2	10	10,2
-25	-20	-11	-4	3	6,7	8,4	9	9,2	9,2
-30	-22	-10	0	6,5	8,6	8,6	8,2	7,7	7,5
-34,6	-24	-6,6	6	10	9,5	7,7	6,3	5,4	5
-34	-25	-2	15	11,6	7,8	5,2	3,4	2,4	2
-4,7	-3,3	1,5	12,4	7,2	3,8	1,3	0	0	-1
		52	8	3,8	-4,6	-2,4	-3,3	-3,6	-3,7
		57	24	36	-2,7	-4,7	-5,2	-5,3	-5,3

Méthode des éléments finis

-2,6	3,8	-2,2	-2,6	-2,9	-2,3	-1,6	-2,2	-2,6	0
-5,3	-3,6	-2,5	-4,6	-3,6	-1	-2,6	-3,6	0	2,6
-5	-6	-3,7	-6,5	-4,6	0,2	-2,7	0	3,6	2,2
-9,2	-7,9	-5,8	-9,3	-2,4	-3,5	0	2,7	2,6	1
-18	-6,4	-9,7	-6,4	-2,3	0	3,6	-0,2	0	1,3
20,9	-11,8	-14,7	-5,2	0	2,3	2,4	3	3,67	2,4
-4,7	-27	-11,8	0	5,7	6,4	8,3	6,5	3,6	2,6
-90	-76,7	7,7	11,8	14,7	8,7	5,8	2,7	2,5	2,2
		92	25,9	10,7	6,4	6,8	6	3,6	3,8
		87	46	19,9	16,9	9,2	5	5,3	2,2

Méthode proposée

Contrainte tangentielle nxy

-0,2	-0,6	-0,8	-0,8	-0,5	-0,2	0	0,1	0,1	0
-0,8	-2,1	-2,6	-2,4	-1,5	-0,5	0,2	0,5	0,5	0,2
-1,4	-3,7	-4,4	-3,6	-1,9	-0,1	0,1	1,3	1	0,4
-2,1	-5,3	-5,9	-4	-1	1,4	2,7	2,7	1,9	7
-2,7	-6,6	-6,6	-2,7	1,8	4,7	5,5	4,7	3	1
-2,5	-5,6	-4,4	2,2	8,5	9,9	8,8	6,7	4,16	1,4
7,2	4,9	3,8	16,8	17,7	14,9	11,5	8,2	4,8	1,6
2,1	64,7	41,8	30	21,6	16,4	11,9	7,8	4,6	1,5
		-1,8	12	15	12	8,9	5,8	3,3	1
		-6,8	-0,8	5	4,5	3,3	2,6	1,2	0,4

Méthode des éléments finis

0,8	0,1	-0,6	0,7	0,8	0,2	0,2	0,7	1,8	1,1
-1,5	1	0	-0,7	0,9	1,7	1,4	3	1	0
-2,3	1,2	-2,7	1,2	2,5	3,7	4,5	1,5	-1	0
6	-2,3	0	5,5	3,4	3	0	-2,3	-0,4	-0,4
0,2	-0,8	6	5,8	2,6	-0,4	-0,1	0,7	-0,5	-0,1
3,7	6,5	4,3	6,6	16	3,8	5,3	0,5	-0,3	-0,2
12,6	26,8	14,9	9,1	3	4,2	-1,9	-2,2	-0,5	-0,2
10	65	23	4,5	8,4	-0,6	0	2,6	1,6	0,7
		11	2,3	5	5,2	3,3	2,3	-0,3	-0
		12	4,6	0,6	3,4	-1,8	0,7	-1,2	-0,8

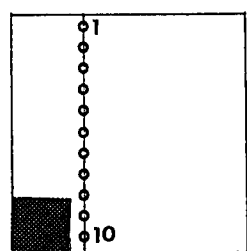
Méthode proposée

Visualisation graphique des résultats.

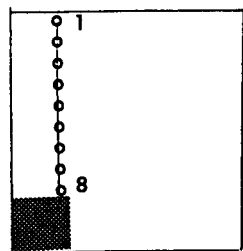
Sur les graphiques des pages suivantes sont portées les variations des contraintes normales n_x s'exerçant en des points situés sur une droite perpendiculaire à l'axe des x . Voir croquis et graphiques a et b.

Pour les points situés sur une droite parallèle à Ox (certains sont communs avec les précédents) afin d'avoir une représentation graphique plus lisible, les contraintes n_x sont tracées perpendiculairement à leur direction réelle. Voir croquis et graphiques c et d.

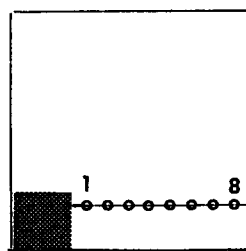
On obtient avec la méthode proposée une majoration des contraintes plus importante pour les points jouxtant le noyau central qu'avec celle des éléments finis. Cette différence est très certainement à mettre au compte de ce qui a été noté précédemment à savoir que pour les zones à fort gradient la méthode des éléments finis donne une moyenne au centre du carré qui peut être très différente de la valeur réelle au point considéré.



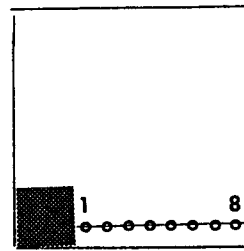
- a -



- b -



- c -

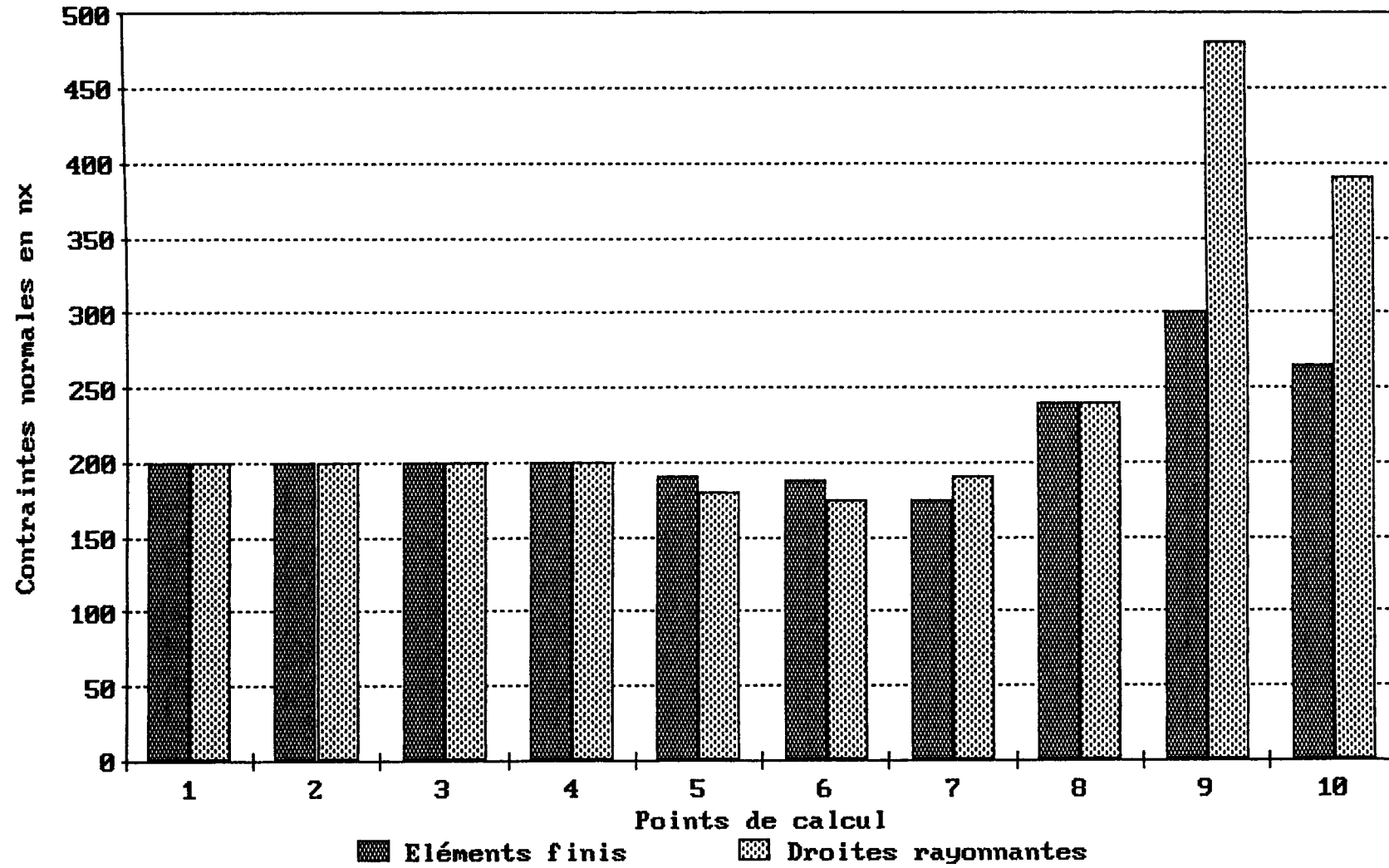


- d -

Les croquis ci-dessus indiquent le repérage des points de calcul, représentés sur le quart supérieur droit de la plaque.

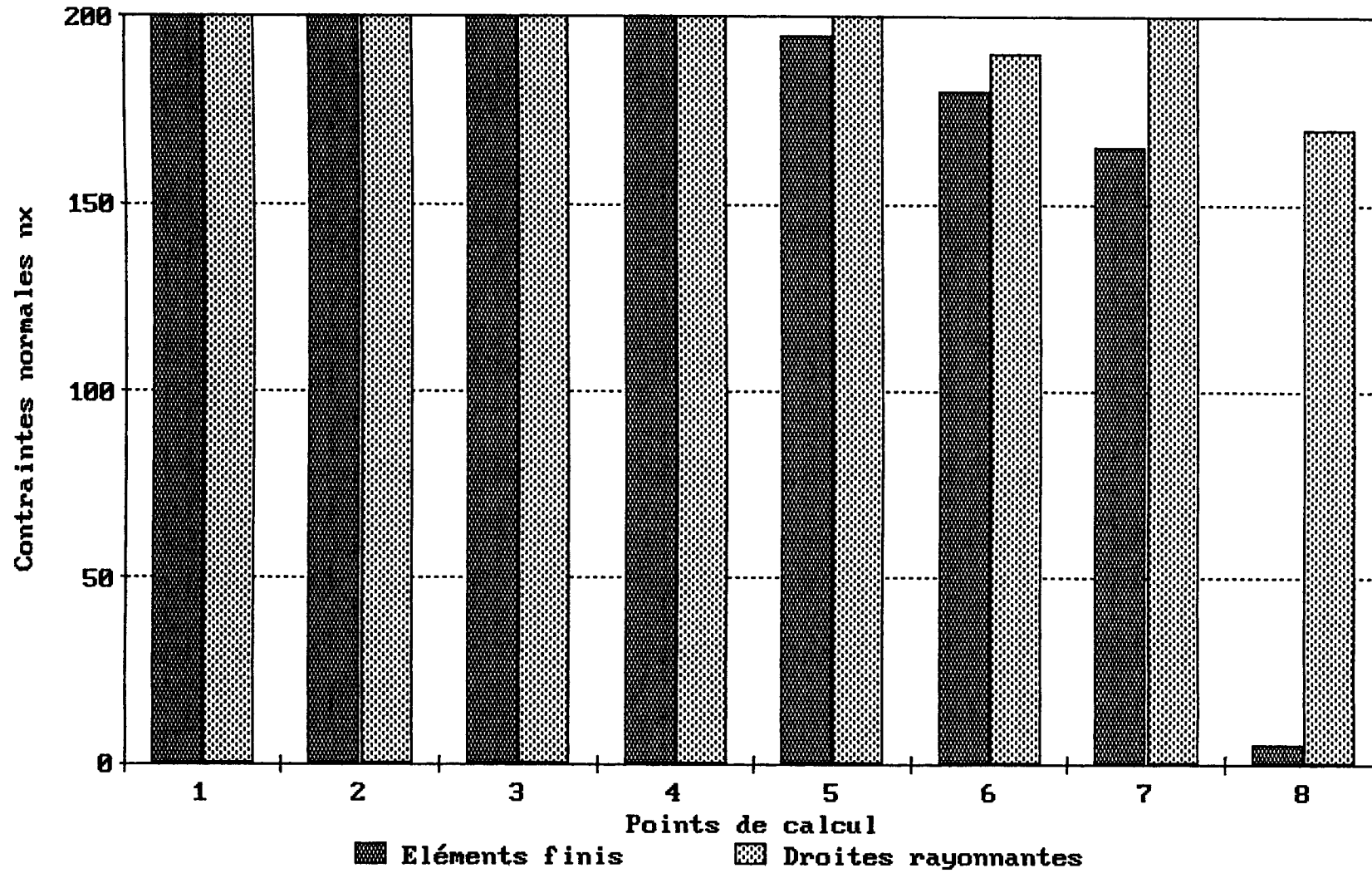
COMPARAISON DES METHODES

figure a



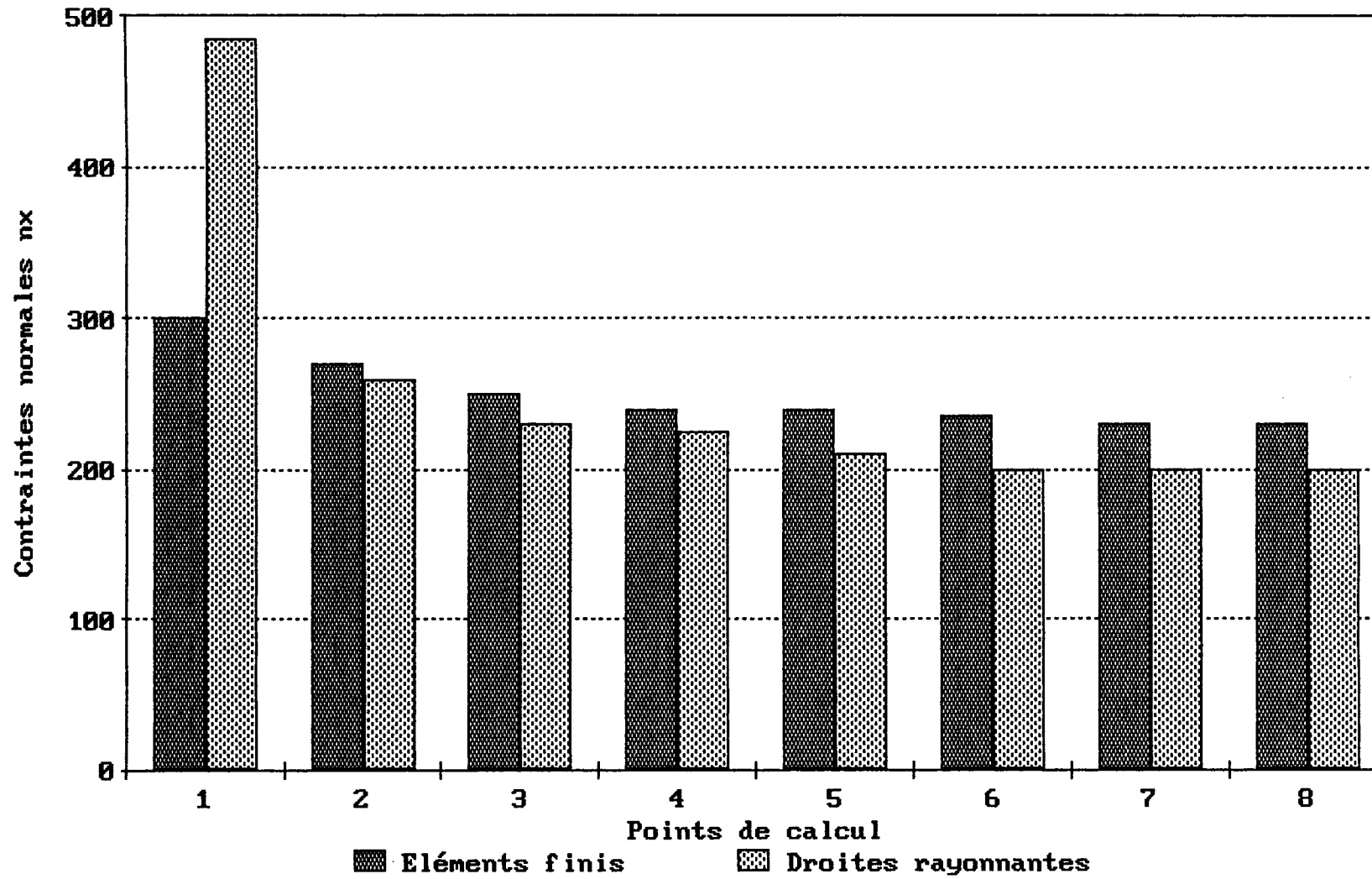
COMPARAISON DES METHODES

figure b



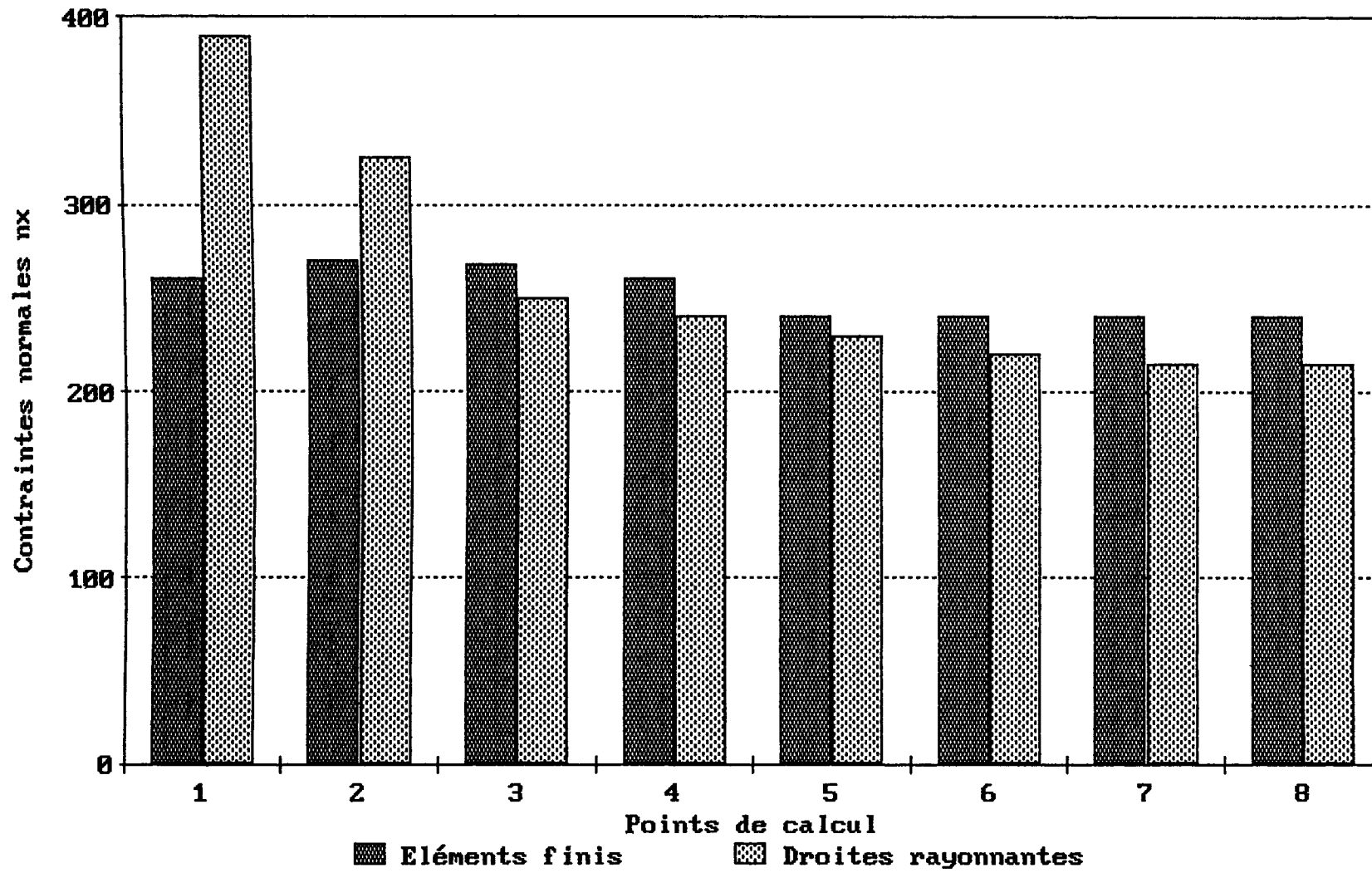
COMPARAISON DES METHODES

Figure c



COMPARAISON DES METHODES

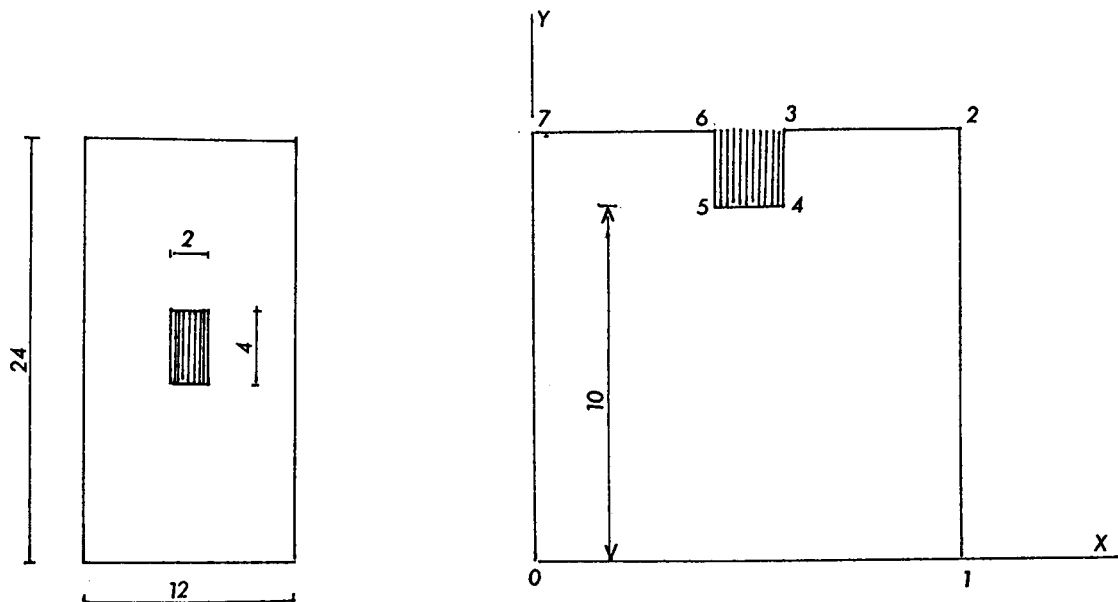
figure d



8 AUTRE APPLICATION

8 Autre application

Effet de poinçonnement dans un matériau composite



L'étude porte sur une plaque (matrice) comportant au centre un élément constitué d'un matériau différent de celui de la plaque.

Dans l'exemple retenu on suppose que le noyau central, subit par rapport à la matrice, une dilatation différentielle, suivant la direction longitudinale Oy .

Cette dilatation entraîne une compression du noyau central, qui se traduit par une répartition de contraintes normales sous les bases, répartition que l'on se propose de déterminer.

Par raison de symétrie l'étude est effectuée sur une demi-plaque, la base du noyau central est représentée par le côté 4.5 du contour.

L'ordonnée du côté 4.5 est égale à 10

Les points de calculs retenus sous la base ont une ordonnée égale à 9.9, ils sont donc très près de cette base par rapport aux côtés.

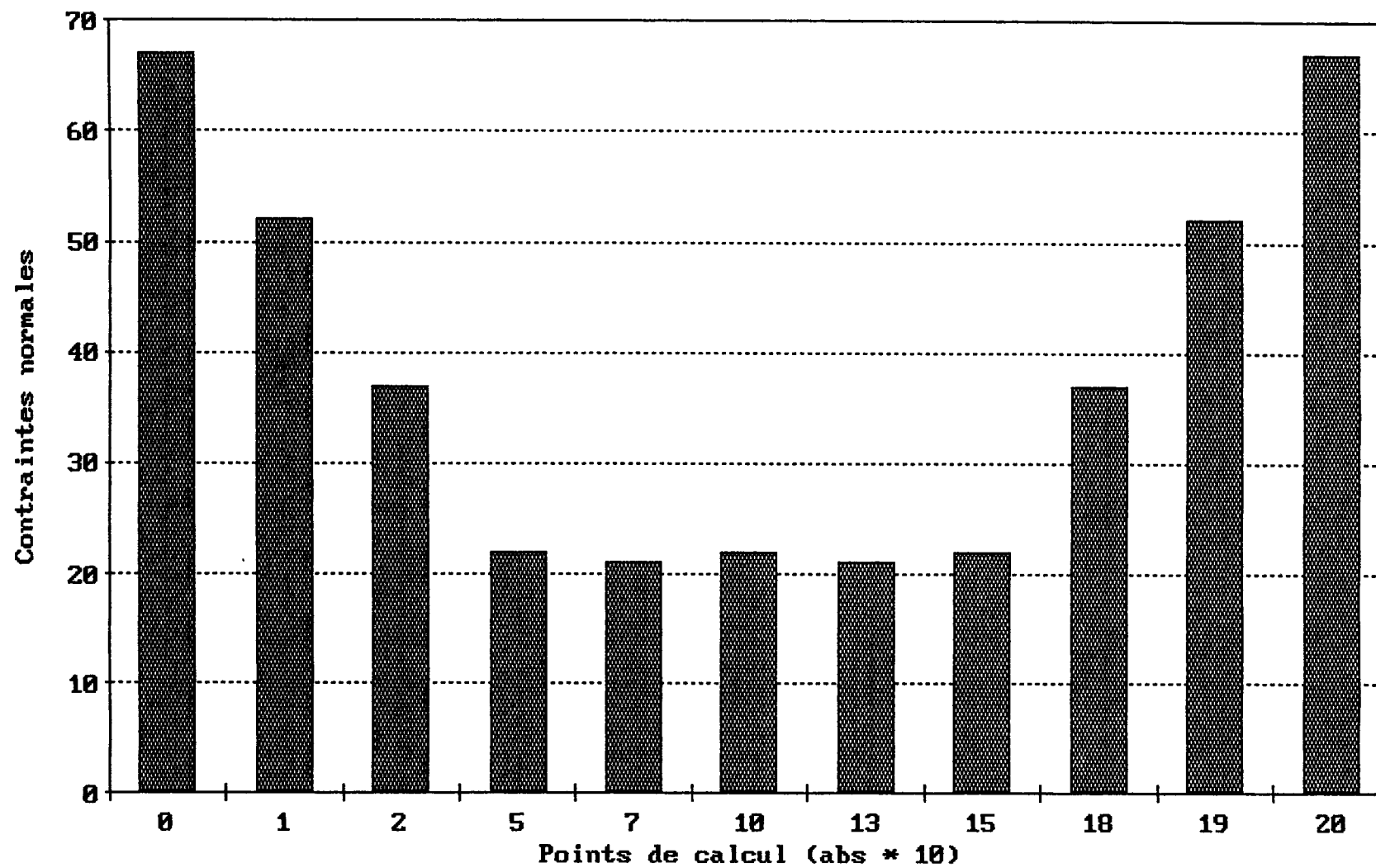
Dans la méthode proposée les variations du pas de maillage suivant l'axe des x et l'utilisation des fenêtres de sorties permettent d'obtenir les contraintes d'une manière continue sous la base de l'élément central.

Cette variation est représentée page suivante.

Le déplacement imposé à la base est égal à 0.001 pour une longueur de 4 soit un déplacement relatif de $1/4000$.

Repartition des contraintes

sous la base du noyau central



9 CONCLUSION

CONCLUSION

Comme nous l'avons développé dans le présent rapport, la méthode proposée, dite des droites rayonnantes, est basée sur la détermination d'un champ de déplacements à l'intérieur d'un matériau, composite ou non.

La détermination de ce champ impose la connaissance à priori des déplacements de tous les points du contour. Cette condition, il convient de la noter, est évidemment restrictif car elle ne permet pas d'aborder les cas de figures pour lesquels seules les forces concentrées ou réparties sont imposées en des points du contour. D'autres méthodes, et en particulier, celles basées sur une modélisation en éléments fixés, permettent d'imposer l'une ou l'autre des conditions sur le contour : forces ou déplacement.

Un développement de notre méthode devrait permettre d'accéder à ces deux conditions mais il n'a pas été envisagé pour l'instant; les cas à étudier dans la formulation actuelle restant très nombreux.

Les tests effectués sur des exemples, dont certains sont donnés dans le rapport, ont permis de mettre en évidence les points particulièrement performants de la méthode, à savoir :

1) L'informalisation nécessite un faible encombrement en mémoire centrale, ceci est dû à l'absence d'une modélisation par maillage qui entraîne, lorsque c'est le cas, un stockage permanent de toutes les caractéristiques de chacun des éléments.<
(Dans la méthode proposée, on utilise un maillage pour l'exploration du domaine, ce qui est totalement différent et ne représente pas une modélisation véritable). Dans sa forme actuelle l'encombrement mémoire à proprement dit est restreint :
Programme 70 Koctets
Données permanentes - quelques Koctets

2) Rapidité de calculs
exemple : Plaque carrée avec noyau central
500 points de mesures
Temps de calcul - 50 secondes

3) Possibilité en adaptant les pas d'exploration du domaine, de déterminer les contraintes en des points très précis et d'obtenir ainsi les variations de ces contraintes d'une manière quasi continue.

Une amélioration immédiate permettant de calculer les contraintes en un point quelconque au choix de l'utilisateur est prévue.

Ainsi que nous l'avions signalé dans notre rapport d'avancement des travaux de recherches du mois de Mars, la mise en oeuvre mathématique et informatique des hypothèses servant de bases théoriques à la méthode, nécessitait des formulations très précises de tous les paramètres entrant dans le calcul des déplacements.

L'écriture et la mise au point d'un nouveau programme répondant rigoureusement à toutes ces exigences a été effectué dans le temps imparti et avec les moyens mis à notre disposition. Plusieurs des développements déjà cités dans le rapport permettraient de rendre ce programme plus convivial et attractif avec une orientation pédagogique.

Nous souhaitons très vivement pouvoir d'une part exploiter le programme dans sa forme actuelle pour effectuer des études comparatives, sur des matériaux composites, avec des chercheurs utilisant d'autres méthodes et d'autre part poursuivre son développement.

Programme P.L.Q

Ce programme est entièrement écrit en langage C. Il a été compilé avec la version 5.0 du compilateur de Microsoft. Il utilise un coprocesseur arithmétique de type 80X87.

```

/*****
/* Calcul d'une plaque .....*/
/*                               */
/* méthode chomarat - savel    */
/*                               */
/* plq.c                        */
*****/

#define MAXSOM 20                /* maximum de sommets */
#define MAXCOTE 20              /* maximum de cotés   */
#define MAXPT 50                /* maximum de points de calcul sur chaque axe */

#define CR '\r'
#define ESC '\033'
#define NORMAL 7
#define INVERSE 70

#include "stdio.h"
#include "math.h"
#include "dos.h"
#include "misc.h"
#include "stdlib.h"
#include "bios.h"
#include "ibmkeys.h"
#include "string.h"

char brk[3] = {CR, ESC};
double a[MAXCOTE], b[MAXCOTE], c[MAXCOTE], x[MAXSOM], y[MAXSOM];
double uc[MAXCOTE], uc1[MAXCOTE], vc[MAXCOTE], vc1[MAXCOTE];
double um[MAXPT][MAXPT], vm[MAXPT][MAXPT], sx[MAXPT][MAXPT], sy[MAXPT][MAXPT], txy[MAXPT][MAXPT];
int val[MAXPT][MAXPT];
int ndeb[MAXCOTE], nfin[MAXCOTE];
int nbcote;

extern void main(void);
extern int pt_plq(double xp, double yp);
extern int dd_cote(double xp, double yp, double u, double v, double ap, double bp, double cp, int ncote, double xm, double ym);
extern void dep(int, double, double, double *, double *);
extern int cd2p(double x1, double y1, double x2, double y2, double *a, double *b, double *c);
extern int interd(double a0, double b0, double c0, double a1, double b1, double c1, double *x, double *y);
extern int pt_cote(double xp, double yp, int ncote);
extern void pprox(double, double, double, double, double *, double *, double *);

```



```

void main()
{
char s[20], cflag;
int i, j, m, n, saut;
int nbs, i1, i2, nray, k1, k2;
int coef[MAXCOTE];
double p, q, nu, E, delta, n1, n2, beta;
double xp[MAXCOTE], yp[MAXCOTE], xmin, xmax, ymin, ymax;
double x1, y1, x2, y2, xm, ym, eps, epsx, epsy, alpha, pi;
double d2, d1, u1, v1, u2, v2, l;
double aa, bb, cc;
double dist[MAXCOTE];
double ux, vx, uy, vy, ex, ey, gxy;
double p1, p2;
double xdeb, xfin, ydeb, yfin;
FILE *fp;

pi = 3.1416;
effecr(0, 24, 0, 79, 0x1F);
atsay(10, 10, "nom du fichier : ");
s[0] = '\0';
if (edt1s(s, 10, 27, 19, NORMAL, NORMAL, brk) == ESC)
    exit(0);

if ((fp = fopen(s, "r")) == NULL)
    exit(0);

atsay(12, 10, "Calculs en Contraintes ou en Déplacements (C/D) : ");
if((cflag = repon('D', "cCd\033", NORMAL, NORMAL, 1)) == ESC)
    exit(0);

fprintf(stdprn, "Nom du fichier : %s\n\n", s);
fscanf(fp, "%lf%lf%d%lf%lf", &p, &q, &nray, &nu, &E);
fprintf(stdprn, "p = %5.3f q = %5.3f nray = %2d nu = %5.2f E = %-10.1f\n\n", p, q, nray,
nu, E);
fscanf(fp, "%d", &nbs);
fprintf(stdprn, "nombre de sommets : %2d\n\n", nbs);
fprintf(stdprn, "coordonnées des sommets :\n\n");
fprintf(stdprn, " Numéro      X          Y\n\n");
for (i = 0; i < nbs; i++)
{
    fscanf(fp, "%lf%lf", &x[i], &y[i]);
    fprintf(stdprn, " %2d      %7.4f %7.4f\n", i, x[i], y[i]);
}
fscanf(fp, "%d", &nbcote);
fprintf(stdprn, "\n\nnombre de cotés : %2d\n\n", nbcote);
fprintf(stdprn, " Conditions imposées sur le contour\n\n");
fprintf(stdprn, " début      fin      u0      u1      v0      v1\n\n");
for (i = 0; i < nbcote; i++)
{
    fscanf(fp, "%d%d%lf%lf%lf%lf", &ndeb[i], &nfin[i], &uc0[i], &uc1[i], &vc0[i], &vc1[i]);
    fprintf(stdprn, " %2d      %2d      %7.4f %7.4f %7.4f %7.4f\n", ndeb[i], nfin[i], uc0[i],
uc1[i], vc0[i], vc1[i]);
}
}

```

```

epsx = 0.01 * p;
epsy = 0.01 * q;
eps = (epsx + epsy) / 2;

/* recherche de l'enveloppe */

xmin = xmax = x[0];
ymin = ymax = y[0];
for (i = 1; i < nbs; i++)
{
    if (x[i] < xmin)
        xmin = x[i];
    else if (x[i] > xmax)
        xmax = x[i];
    if (y[i] < ymin)
        ymin = y[i];
    else if (y[i] > ymax)
        ymax = y[i];
}

atsay(14, 10, "Sortie en x de: ");
sprintf(s, "%7.4f", xmin);
if(edtn(s, 14, 26, 7, 4, NORMAL, NORMAL, brk, 1, 0, 1) == ESC)
    exit(1);
xdeb = atof(s);
atsay(14, 36, "à: ");
sprintf(s, "%7.4f", xmax);
if(edtn(s, 14, 40, 7, 4, NORMAL, NORMAL, brk, 1, 0, 1) == ESC)
    exit(1);
xfin = atof(s);

atsay(16, 10, "Sortie en y de: ");
sprintf(s, "%7.4f", ymin);
if(edtn(s, 16, 26, 7, 4, NORMAL, NORMAL, brk, 1, 0, 1) == ESC)
    exit(1);
ydeb = atof(s);
atsay(16, 36, "à: ");
sprintf(s, "%7.4f", ymax);
if(edtn(s, 16, 40, 7, 4, NORMAL, NORMAL, brk, 1, 0, 1) == ESC)
    exit(1);
yfin = atof(s);

```

```
/* Coefficients des droites des barres */
```

```
for (i = 0; i < nbcote; i++)  
{  
  i1 = ndeb[i];  
  i2 = nfin[i];  
  x1 = x[i1];  
  x2 = x[i2];  
  y1 = y[i1];  
  y2 = y[i2];  
  a[i] = y2 - y1;  
  b[i] = x1 - x2;  
  c[i] = y1 * x2 - x1 * y2;  
}
```



```

xmin = xmin + p / 2;
ymin = ymin + q / 2;

/* Calcul des déplacements de chaque point */

for (ym = ymin, m = 0; ym < ymax; ym = ym + q, m++)
{
  for (xm = xmin, n = 0; xm < xmax; xm = xm + p, n++)
  {
    if (pt_plq(xm, ym) != 1)
    {
      val[n][m] = 0;
      continue;
    }

    curset(20, 20, 0);
    printf("y = %7.4f    x = %7.4f", ym, xm);

    val[n][m] = 1;
    um[n][m] = vm[n][m] = 0.;
    for (alpha = (-pi / 2); alpha < pi / 2 - 0.01; alpha = alpha + pi / nray)
    {
      aa = -sin(alpha);
      bb = cos(alpha);
      cc = -aa * xm - bb * ym;

/* Détermination des points d'intersection sur les cotés */

      for (i = 0; i < nbcote; i++)
        coef[i] = interd(aa, bb, cc, a[i], b[i], c[i], &xp[i], &yp[i]);

/* élimination des points hors des cotés */

      for (i = 0; i < nbcote; i++)
      {
        if (coef[i] == 0)
          continue;
        if (pt_cote(xp[i], yp[i], i) < 0)
          coef[i] = 0;
      }

/* élimination des points doubles */

      for (i = 0; i < nbcote; i++)
      {
        if (coef[i] == 0)
          continue;
        for (j = 0; j < nbcote; j++)
        {
          if (j == i || coef[j] == 0)
            continue;
          if (xp[i] < xp[j] + eps && xp[i] > xp[j] - eps &&
              yp[i] < yp[j] + eps && yp[i] > yp[j] - eps)
            coef[j] = 0;
        }
      }

```

```

/* calcul des distances */

    for (i = 0; i < nbcote; i++)
    {
        if (coef[i] == 0)
            continue;
        dist[i] = sqrt((xp[i] - xm) * (xp[i] - xm) + (yp[i] - ym) * (yp[i] - ym));
        if (bb != 0.)
        {
            if (xm > xp[i])
                dist[i] = -dist[i];
        }
        else
        {
            if (ym > yp[i])
                dist[i] = -dist[i];
        }
    }

/* recherche du point le plus proche de chaque coté */

    d2 = 1.E38;
    d1 = -1.E38;
    for (i = 0; i < nbcote; i++)
    {
        if (coef[i] == 0)
            continue;
        if (dist[i] < 0. && dist[i] > d1)
        {
            k1 = i;
            d1 = dist[i];
        }
        else if (dist[i] > 0. && dist[i] < d2)
        {
            k2 = i;
            d2 = dist[i];
        }
    }

/* déplacements élémentaires du point courant et sommation */

    dep(k1, xp[k1], yp[k1], &u1, &v1);
    dep(k2, xp[k2], yp[k2], &u2, &v2);
    d1 = -d1;
    l = d1 + d2;
    um[n][m] = um[n][m] + (d1 * u2 + d2 * u1) / l;
    vm[n][m] = vm[n][m] + (d1 * v2 + d2 * v1) / l;
}

/* déplacements du point courant */
    um[n][m] = um[n][m] / nray;
    vm[n][m] = vm[n][m] / nray;
}

```



```

/* dérivée en y */

if (j > 0 && j < m - 1 && val[i][j - 1] == 1 && val[i][j + 1] == 1)
{
    /* cas général */
    uy = (um[i][j + 1] - um[i][j - 1]) / (2 * q);
    vy = (vm[i][j + 1] - vm[i][j - 1]) / (2 * q);
}
else
{
    if (j == 0 || val[i][j - 1] == 0)
    {
        /* calcul de p1, u1 et v1 */
        pprox(xm, ym, 0., -1., &p1, &u1, &v1);
    }
    else
    {
        p1 = q;
        u1 = um[i][j - 1];
        v1 = vm[i][j - 1];
    }

    if (j == m - 1 || val[i][j + 1] == 0)
    {
        /* calcul de p2, u2 et v2 */
        pprox(xm, ym, 0., 1., &p2, &u2, &v2);
    }
    else
    {
        p2 = q;
        u2 = um[i][j + 1];
        v2 = vm[i][j + 1];
    }

    uy = -p2 * u1 / (p1 * (p1 + p2))
        - (p1 - p2) * um[i][j] / (p1 * p2)
        + p1 * u2 / ((p1 + p2) * p2);
    vy = -p2 * v1 / (p1 * (p1 + p2))
        - (p1 - p2) * vm[i][j] / (p1 * p2)
        + p1 * v2 / ((p1 + p2) * p2);
}

```

```

ex = ux;
ey = vy;
gxy = uy + vx;

if(cflag == 'C')
{
delta = E / (1 - nu * nu);
sx[i][j] = delta * (ex + nu * ey);
sy[i][j] = delta * (ey + nu * ex);
}
else
{
delta = E / ((1 + nu) * (1 - 2 * nu));
sx[i][j] = delta * ((1 - nu) * ex + nu * ey);
sy[i][j] = delta * ((1 - nu) * ey + nu * ex);
}
txy[i][j] = E * gxy / (2 * (1 + nu));

delta = (sx[i][j] - sy[i][j]) / 2;
delta = sqrt(delta * delta + txy[i][j] * txy[i][j]);
n1 = (sx[i][j] + sy[i][j]) / 2 + delta;
n2 = (sx[i][j] + sy[i][j]) / 2 - delta;

beta = atan(2 * txy[i][j] / (sx[i][j] - sy[i][j])) / 2;
beta = beta * 180. / 3.141592;
if(sx[i][j] < sy[i][j])
beta = beta + 90.;

if(xm >= xdeb && xm <= xfin && ym >= ydeb && ym <= yfin)
{
saut = 1;
fprintf(stdprn, "%5.2f%7.2f%9.5f%9.5f%9.2f%9.2f%8.2f%9.2f%9.2f%5.0f\n",
xm, ym, um[i][j], vm[i][j], sx[i][j], sy[i][j], txy[i][j], n1, n2, beta);
}
}
if(saut != 0)
fprintf(stdprn, "\n");
}
fprintf(stdprn, "\014");
effecr(0, 24, 0, 79, 0x1F);
}

```

```

    /***/
    /* intersection d'une demi droite */
    /*   et d'un coté d'une plaque   */
    /***/

int dd_cote(xp, yp, u, v, ap, bp, cp, ncote, xm, ym)
int ncote;
double xp, yp, u, v, ap, bp, cp, *xm, *ym;
{
    /* dd_cote */

    /* retour 0 si pas coupure (coupure à l'extérieur du coté */
    /*         1 si coupure à l'intérieur du coté */
    /*         2 si coupure à un sommet */
    /*         3 si demi droite et coté // */
    /*         4 le point est sur le coté */

    int k, i1, i2;
    double as, bs, cs, xd, yd, xf, yf;

    as = a[ncote];
    bs = b[ncote];
    cs = c[ncote];
    i1 = ndeb[ncote];
    i2 = nfin[ncote];
    xd = x[i1];
    yd = y[i1];
    xf = x[i2];
    yf = y[i2];

    /* le point est sur le coté */
    if (as * xp + bs * yp + cs == 0.)
    {
        if (pt_cote(xp, yp, ncote) != (-1))
            return (4);
        else
            return (0);
    }

    k = interd(ap, bp, cp, as, bs, cs, xm, ym);

    if (k == 0)          /* demi droite et cote // */
        return (3);

    /* test si le point de coupure est sur la demi droite */
    if (u != 0. && (*xm - xp) / u < 0.)
        return (0);
    if (v != 0. && (*ym - yp) / v < 0.)
        return (0);

    k = pt_cote(*xm, *ym, ncote);

    /* le point de coupure est à un sommet */
    if (k == (-1))
        return (0);
    if (k == 0)
        return (2);
    return (1);
}
    /* dd_cote */

```

```

    /***/
    /* coefficient d'une droite */
    /* donnée par deux points */
    /***/

int cd2p(x1, y1, x2, y2, a, b, c)
double x1, x2, y1, y2, *a, *b, *c;
{
    /* cd2p */

    /* retour 0 si droite définie */
    /*      1 sinon */

    *a = y2 - y1;
    *b = x1 - x2;
    *c = y1 * x2 - x1 * y2;
    if (*a == 0. && *b == 0.)
        return (1);
    else
        return (0);
}
    /* cd2p */

    /***/
    /* déplacements des cotés */
    /***/

void dep(i, xp, yp, u, v)
int i;
double xp, yp, *u, *v;
{
    int i1, i2;
    double xd, xf, yd, yf, l, a11, a12, a21, a22, gx, gu, gv;

    i1 = ndeb[i];
    i2 = nfin[i];
    xd = x[i1];
    xf = x[i2];
    yd = y[i1];
    yf = y[i2];
    l = hypot(xf - xd, yf - yd);

    a11 = (xf - xd) / l;
    a21 = (yf - yd) / l;
    a12 = -a21;
    a22 = a11;
    gx = hypot(xp - xd, yp - yd);
    gu = uc0[i] + uc1[i] * gx;
    gv = vc0[i] + vc1[i] * gx;
    *u = a11 * gu + a12 * gv;
    *v = a21 * gu + a22 * gv;
}

```



```

/*****/
/* intersection de deux droites */
/*****/

int interd(a0, b0, c0, a1, b1, c1, x, y)
double a0, b0, c0, a1, b1, c1, *x, *y;
{
    /* interd */

    /* retour 0 si les 2 droites // */
    /* retour 1 sinon */

    double delta;
    delta = a0 * b1 - a1 * b0;
    if (delta > -0.001 && delta < 0.001)
        return (0);
    *x = (c1 * b0 - c0 * b1) / delta;
    *y = (a1 * c0 - a0 * c1) / delta;
    return (1);
}

/*****/
/* détermination du point le plus proche */
/*****/

void pprox(xm, ym, u, v, pp, up, vp)
double xm, ym, u, v, *pp, *up, *vp;
{
    /* pprox */
    int i, k, ncote;
    double d, ap, bp, cp, xp, yp, xprox, yprox;

    *pp = 1.E38;
    cd2p(xm, ym, xm + u, ym + v, &ap, &bp, &cp);

    for (i = 0; i < nbcote; i++)
    {
        if ((k = dd_cote(xm, ym, u, v, ap, bp, cp, i, &xp, &yp)) != 1 && k != 2)
            continue;
        d = hypot(xp - xm, yp - ym);
        if (d < *pp)
        {
            *pp = d;
            ncote = i;
            xprox = xp;
            yprox = yp;
        }
    }
    dep(ncote, xprox, yprox, up, vp);
}
/* pprox */

```

```

/*****
/* test si un point est sur un segment */
/* sachant le point sur le support */
*****/

int pt_cote(xp, yp, ncote)
int ncote;
double xp, yp;
{
    /* pt_cote */
    int i1, i2;

    /* retour -1 à l'extérieur */
    /*          0 sur un sommet */
    /*          1 à l'intérieur */

    double sauve, xd, yd, xf, yf;

    i1 = ndeb[ncote];
    i2 = nfin[ncote];
    xd = x[i1];
    yd = y[i1];
    xf = x[i2];
    yf = y[i2];

    if (xp == xd && yp == yd) /* le point sur un sommet */
        return (0);
    if (xp == xf && yp == yf)
        return (0);

    if (xd == xf) /* le cote est selon Oy */
    {
        if (yd > yf)
        {
            sauve = yd;
            yd = yf;
            yf = sauve;
        }
        if (yp < yd || yp > yf)
            return (-1);
        return (1);
    }

    if (xf < xd) /* cas général */
    {
        sauve = xf;
        xf = xd;
        xd = sauve;
    }
    if (xp < xd || xp > xf)
        return (-1);
    return (1);
}
/* pt_cote */

```

```

/*****
/* position relative d'un point et d'une plaque */
*****/

int pt_plq(xp, yp)
double xp, yp;
{

/* retour 1 point à l'intérieur */
/*      0      sur le contour */
/*     -1      à l'extérieur */

int nbinter = 0, i, k;
double u, v, ap, bp, cp, xm, ym;

u = v = 1.;
cd2p(xp, yp, xp + u, yp + u, &ap, &bp, &cp);
for (i = 0; i < nbcote; i++)
{
k = dd_cote(xp, yp, u, v, ap, bp, cp, i, &xm, &ym);
if (k == 4)
return (0);
if (k > 1)
{
u = u / 2.;
nbinter = 0;
i = (-1);
cd2p(xp, yp, xp + u, yp + v, &ap, &bp, &cp);
continue;
}
nbinter = nbinter + k;
}
k = nbinter % 2;
if (k == 0)
return (-1);
else
return (1);
}

```