

Université des Sciences et de la Technologie d'Alger

47/82

207

**ECOLE NATIONALE POLYTECHNIQUE**

DEPARTEMENT D'ELECTRONIQUE ET D'ELECTROTECHNIQUE

FILIERE D'INGENIEUR EN ELECTRONIQUE

المدرسة الوطنية للعلوم الهندسية  
**PROJET DE FIN D'ETUDES**  
ECOLE NATIONALE POLYTECHNIQUE  
BIBLIOTHEQUE

Sujet : Choix du maillage pour l'uniformisation de l'erreur  
due à la résolution des équations dérivées partielles

Proposé par :

**J - P GAUTHIER**

Docteur Ingénieur

Réalisé par :

**SEBIHI Bachir**

**RIHANI Nacer**

Janvier 1982

Université des Sciences et de la Technologie d'Alger

**ECOLE NATIONALE POLYTECHNIQUE**

DEPARTEMENT D'ELECTRONIQUE ET D'ELECTROTECHNIQUE

FILIERE D'INGENIEUR EN ELECTRONIQUE

**PROJET DE FIN D'ETUDES**

Sujet : Choix du maillage pour l'uniformisation de l'erreur  
due à la résolution des équations dérivées partielles

Proposé par :

**J - P GAUTHIER**

Docteur Ingénieur

Réalisé par :

**SEBIHI Bachir**

**RIHANI Nacer**

Janvier 1982

DEDICACES

A CHAQUE MEMBRE DE LA FAMILLE ET A TOUS LES AMIS

SEBIHI BACHIR

A MES PARENTS

A MES AMIS

RIHANI NACER

R E M E R C I E M E N T S

Nous tenons à remercier plus particulièrement  
notre promoteur pour nous avoir guidé tout le  
long de notre travail, nous remercions également  
les membres du CENTRE NATIONALE DU TRAITEMENT  
DE L'INFORMATION qui nous ont facilité l'accès  
à l'ordinateur.

I N T R O D U C T I O N  
-----oooOooo-----

Le travail que nous allons vous présenter est une amélioration des résultats relatifs à la résolution des équations aux dérivées partielles. Le problème consiste à chercher la solution en chaque point d'un carre unitaire  $(0,1) \times (0,1)$  par discretisation et avec des conditions aux bords sur chaque cote du carre.

On constate que l'erreur devient de plus en plus importante au fur et à mesure que l'on s'approche du centre du carre et cela quelque soit la méthode de résolution adoptée (GAUS, JACOBI, GAUSS-SEIDEL). Que ce soit avec un maillage régulier ou irrégulier, les résultats ne sont pas satisfaisants bien que dans le 2ème maillage ils sont meilleurs.

Notre but est donc de proposer une méthode qui permet d'avoir une erreur aussi uniforme que possible. Ainsi nous aurons une courbe de l'erreur aussi aplatie que possible.

Vous trouverez dans ce polycopie les détails de cette méthode.

.../...

P L A N

- A - Notions de mathématiques
  - I. Matrices
  - II. Normes vectorielles
  - III. Valeurs et vecteurs propres
  - IV. Méthode de la puissance itérée
  - V. Minimisation: Méthode du gradient
- B - EQUATIONS AUX DERIVEES PARTIELLES
  - I. Forme générale
  - II. Mise sous la forme  $AX=B$
  - III. Remplissage de A et B
- C - UNIFORMISATION DE L'ERREUR
  - I. Cas général de maillage irrégulier
    - I.1- Division en k maillages
    - I.2- Détermination des indices
    - I.3- Distribution des maillages
  - II. Calcul de la matrice d'iteration
  - III. Calcul des normes
  - IV. Calcul des valeurs propres et vecteurs propres.
  - V. Minimisation
- D - Résolution et comparaison
- E - Conclusion

A - NOTIONS DE MATHÉMATIQUESI-MATRICEI-1-Matrice non singulière (ou régulière)

C'est une matrice A dont le déterminant est différent de zéro.

I-2-Matrice non négative

Une matrice A est dite non négative si tous les éléments  $a_{ij}$  sont positifs ou nuls.

I-3-Z. Matrice

Soit A une matrice carrée réelle. On dira que A est une Z. Matrice si tous ses éléments hors-diagonaux sont négatifs ou nuls.

I-4-M. Matrice

C'est cas particulier d'une Z. Matrice qui admet une inverse non négative.

Pour qu'une Z. Matrice soit une M. Matrice il faut et il suffit :

- que tous ses éléments soient strictement positifs (la Matrice de JACOBI associée à cette matrice est donc bien définie et est non négative).

- que le rayon spectral de la matrice J de JACOBI soit inférieur à l'unité  $\rho(J) < 1$ .

On rappelle que le rayon spectral d'une matrice carrée est le maximum de ses valeurs propres.  $\rho(A) = \max_i |\lambda_i|$

I-5-Remarque

Si A est une M. Matrice les méthodes de JACOBI et de GAUSS-SEIDEL pour la résolution d'un système linéaire ayant pour matrice A sont convergentes. La seconde convergeant au moins aussi vite que la première.

I-6-H.Matrice

A étant une matrice complexe (n,n) et N(A) sa matrice réelle définie comme suit:

$$\begin{vmatrix} |a_{11}| & \dots & -|a_{1j}| \\ \vdots & \ddots & \vdots \\ -|a_{ij}| & \dots & -|a_{nn}| \end{vmatrix}$$

A est dite H.Matrice si la Z.Matrice N(A) est une M.Matrice.

Toute M.Matrice est une H.Matrice. On a alors  $N(A)=A$ .

Pour qu'une matrice complexe (n,n) A soit une H.Matrice il faut et il suffit:

- que la diagonal D de A soit non singulière.
- que  $\rho(|J|) < 1$  J étant la matrice de JACOBI associée à A.
- que si A est une H.Matrice, les méthodes de JACOBI et de GAUSS-SEIDEL pour un système linéaire de matrice A sont convergentes.

I-7-Theorie de PERRON FROBENIUSTheoreme 1:(cas general)

Soit A une matrice (n,n) nonnegative alors :

- son rayon spectral est valeur propre positive ou nul de A
- il lui correspond  $W > 0$  dans  $R^n$  tel que  $AW = \rho(A)W$ .
- si  $B \geq A$  alors  $\rho(B) \geq \rho(A)$
- $0 \leq \sup_{\substack{u > 0 \\ u \in R^n}} \min_i \sum_j \frac{a_{ij} u_j}{u_i} \leq \rho(A) \leq \inf_{\substack{u > 0 \\ u \in R^n}} \max_i \sum_j \frac{a_{ij} u_j}{u_i}$
- pour toute matrice B, réelle ou complexe il vient:

$$\rho(B) \leq \rho(|B|)$$

Theoreme 2:(cas irreductible)

Soit  $A=(a_{ij})$  une matrice (n,n) nonnegative et irreductible alors:

- son rayon spectral  $\rho(A)$  est valeur propre simple et positive de A.
- il lui correspond un vecteur propre  $W, W > 0$  dans  $R^n$
- $AW = \rho(A)W$



-si  $B \geq A$  et  $B \neq A$  alors  $\rho(A) < \rho(B)$ .

-on a pour tout  $u >^0$  dans  $\mathbb{R}^n$  :

$$\text{soit } \min_i \sum_j \frac{a_{ij} u_j}{u_i} < \rho(A) < \max_i \sum_j \frac{a_{ij} u_j}{u_i}$$

$$\text{soit } \rho(A) = \sum_j \frac{a_{ij} u_j}{u_i} \quad (i=1,2,\dots,n)$$

comme par exemple pour  $u=w$ . D'où

$$\max_{\substack{u >^0 \\ u \in \mathbb{R}^n}} \min_i \sum_j \frac{a_{ij} u_j}{u_i} = \rho(A) = \min_{\substack{u >^0 \\ u \in \mathbb{R}^n}} \max_i \sum_j \frac{a_{ij} u_j}{u_i}$$

## II-NORMES VECTORIELLES

Soit  $X$  un espace vectoriel sur  $\mathbb{R}^n$ . Toute application  $P$  de  $X$  dans  $\mathbb{R}^k$  vérifiant les axiomes suivants :

- a)  $\forall \lambda \in \mathbb{R}, \forall x \in X \quad P(\lambda x) = |\lambda| P(x)$
- b)  $\forall x, y \in X \quad P(x+y) \leq P(x) + P(y)$
- c)  $P(x) = 0 \implies x = 0$

est appelée norme vectorielle de taille  $k$  sur  $X$ . Si une telle application  $P$  existe,  $X$  sera dit vectoriellement norme par  $P$ .

si  $P$  existe de a) on déduit pour  $\lambda = 0$

$$P(0) = 0 \text{ donc } P(x) = 0 \iff x = 0$$

de a) et b) on déduit :

$$\forall x, y \in X \quad |P(x) - P(y)| \leq P(x-y)$$

Soit  $P_i(n)$  la  $i$ ème composante de  $P(n)$  dans  $\mathbb{R}^k$  ( $i = 1, 2, \dots, k$ ) les deux premiers axiomes montrent que les  $k$  applications  $P_i$  sont des semi-normes sur  $X$ .

Cas où  $X = M_{m,n}(\mathbb{R})$  un espace vectoriel de matrices complexes  $(m,n)$  et pour  $A = (a_{ij})$  dans  $M_{m,n}$

$$P(A) = \begin{pmatrix} P_1(A) \\ \vdots \\ P_m(A) \end{pmatrix} \quad \text{avec } P_i(A) = \sum_{j=1}^n a_{ij} \quad i = (1, 2, \dots, m)$$

norme vectorielle de taille  $m$  sur  $M_{m,n}$  :

$$P(A) = \max_{i=1}^m P_i(A) = S_{\infty}(A)$$

### III- VALEURS ET VECTEURS PROPRES

Soit un système de  $n$  équations linéaires et homogènes du type:

$$AX = \lambda X$$

où  $A$  est une matrice carrée,  $X$  une matrice colonne et  $\lambda$  un scalaire.  
Cela s'écrit aussi :

$$(A - \lambda I) X = 0 \quad \text{ou } I \text{ est la matrice unité d'ordre } n.$$

Un tel système admet toujours la solution  $X = 0$  qui ne présente aucun intérêt. Pour qu'il admette d'autres solutions, il faut que le déterminant des inconnues soit nul d'où:

$$||A - \lambda I|| = 0$$

$$\begin{vmatrix} (a_{11} - \lambda) & a_{12} & \dots & a_{1n} \\ a_{21} & (a_{22} - \lambda) & a_{23} & \dots & a_{2n} \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ a_{n1} & a_{n2} & a_{n3} & \dots & (a_{nn} - \lambda) \end{vmatrix}$$

Le déterminant  $D$  conduit à un polynôme en  $\lambda$  de degré  $n$ .

$$P(\lambda) = (-1)^n (\lambda^n + a_1 \lambda^{n-1} + a_2 \lambda^{n-2} + \dots + a_{n-1} \lambda + a_n)$$

Ce polynôme est appelé caractéristique. Par conséquent seules les racines  $\lambda_i$  de ce polynôme au nombre de  $n$  réelles ou complexes correspondent à des solutions non nulles.

Les racines  $\lambda_i$  de  $P(\lambda)$  sont les valeurs propres de la matrice  $A$ .

A chaque valeur de  $\lambda_i$  correspond un vecteur que nous designons par  $X_i$  et que l'on appelle le vecteur propre de A associe  $\lambda_i$  on a donc

$$AX_i = \lambda_i X_i$$

#### 6 IV - METHODE DE LA PUISSANCE ITEREE

C'est une methode iterative qui permet de calculer les valeurs propres et les vecteurs propres d'une matrice A.

##### Theoreme

Soit une matrice A,  $A \in M_{n,n}(\mathbb{R})$  ayant la propriete suivante:  
A admet la valeur propre  $\lambda_1 > 0$  tel que  $\lambda_1 = \rho(A) = \max |\lambda_i|$  et c'est une valeur simple du polynome caracteristique de A et  $|\lambda_i| < \lambda_1$  ( $i \neq 1$ ) alors si  $X_0$  est un vecteur quelconque non orthogonal a la direction V tel que  $A^t V = \lambda_1 V$  ( $V^t X_0 = 0$ ) et  $\varphi$  une norme de vecteur sur  $\mathbb{R}^n$  l'iteration:

$$X_{n+1} = (AX_n) / \varphi(AX_n)$$

converge pour  $n \rightarrow \infty$  vers le vecteur U tel que  $AU = \lambda_1 U$  et  $\varphi(U) = 1$  et le rapport des deux composantes:

$$\frac{(AX_n)_i}{(X_n)_i} \rightarrow \lambda_1$$

On obtient donc la valeur propre la plus grande en module .

$X_0$  est choisi en general = ( 1,0000.....,0).

$\varphi$  sera la norme du maximum .

$$\varphi(AX_n) = \max_i |(AX_n)_i|$$

Le calcul des autres valeurs propres sera obtenu par deflations:

$$A' = A - \frac{\lambda_1 v_1^t u_1^t}{u_1^t v_1}$$

A' admet comme valeurs propres ( 0,  $\lambda_2, \lambda_3, \dots, \lambda_n$ )

$$\lambda_1 \gg \lambda_2 \dots \dots \dots \gg \lambda_n$$

$$A'' = A' - \frac{\lambda_2 U_2 V_2^t}{U_2^t V_2}$$

$$A^{(n-1)} = A^{(n-2)} - \lambda_{(n-1)} \cdot \frac{U_{(n-1)} V_{(n-1)}^t}{U_{(n-1)}^t V_{(n-1)}}$$

$A^{(n-1)}$  admet comme valeurs propres  $(0, 0, 0, \dots, \lambda_n)$ .

### V - MINIMISATION: METHODE DU GRADIENT

Soit un système de n équations défini comme suit :

$$F_p(x_1, x_2, \dots, x_n) = 0 \quad p = 1, 2, \dots, n$$

considérons la fonction  $\phi = \sum_{p=1}^n F_p^2$   
ou plus généralement

$$\phi = \sum_{i,j} a_{ij} F_i F_j$$

ou les  $a_{ij}$  sont les éléments d'une matrice définie positive.

Le problème revient donc à chercher le minimum de la fonction et pour cela on diminue sa valeur à partir de celle qu'elle prend pour une valeur arbitraire  $x_p(0)$  des inconnues.

Partant d'un point  $X$  de coordonnées  $(x_1, x_2, \dots, x_n)$

ou la fonction  $\phi$  a la valeur  $\phi(X)$  on passe à un autre point:  
 $(X + \mu V)$

ou  $V$  est un vecteur arbitraire que nous préciserons plus loin et  $\mu$  un paramètre. On choisit  $\mu$  de manière que  $\phi(\mu)$  soit la plus petite possible. On ramène donc le problème à  $n$  variables à un problème à un seul paramètre.

En développant  $\phi(X + \mu V)$  en série de TAYLOR autour de  $X$  on a : en se limitant aux termes en  $\mu^2$ , en écrivant :

$$\phi(X + \mu V) = \phi(X) + \mu \sum_p V_p \frac{\partial \phi}{\partial X_p} + \frac{\mu^2}{2} \sum_p \sum_q V_p V_q \frac{\partial^2 \phi}{\partial X_p \partial X_q}$$

$$\frac{d\phi}{d\mu} = 0$$

$$\mu = \frac{-\sum_p V_p \frac{\partial \phi}{\partial X_p}}{\sum_p \sum_q V_p V_q \frac{\partial^2 \phi}{\partial X_p \partial X_q}}$$

Le vecteur  $V$  est choisi arbitrairement. On peut prendre par exemple l'un des vecteurs unitaires  $E_k$  des axes de coordonnées  $V = E_k$ .

### METHODE DU GRADIENT

Si  $V = \text{grad}(\phi)$  ou  $V_k = \nabla_k \phi$  alors on obtient la methode du gradient ou de plus grande pente.

### Diminution dans la direction de plus grande pente

Partant de  $(x_0, y_0)$  la direction de plus grande pente est donnee par le vecteur gradient change de signe  $-\left(\frac{\partial \phi}{\partial x}\right)_k - \left(\frac{\partial \phi}{\partial y}\right)_k$ . On peut chercher un nouveau point dans cette direction en se deplacant d'une longueur  $h$ ; on a donc

$$\begin{aligned} x_{(k+1)} &= x_k - h \frac{\partial \phi}{\partial x}(k) \\ y_{(k+1)} &= y_k - h \frac{\partial \phi}{\partial y}(k) \end{aligned}$$

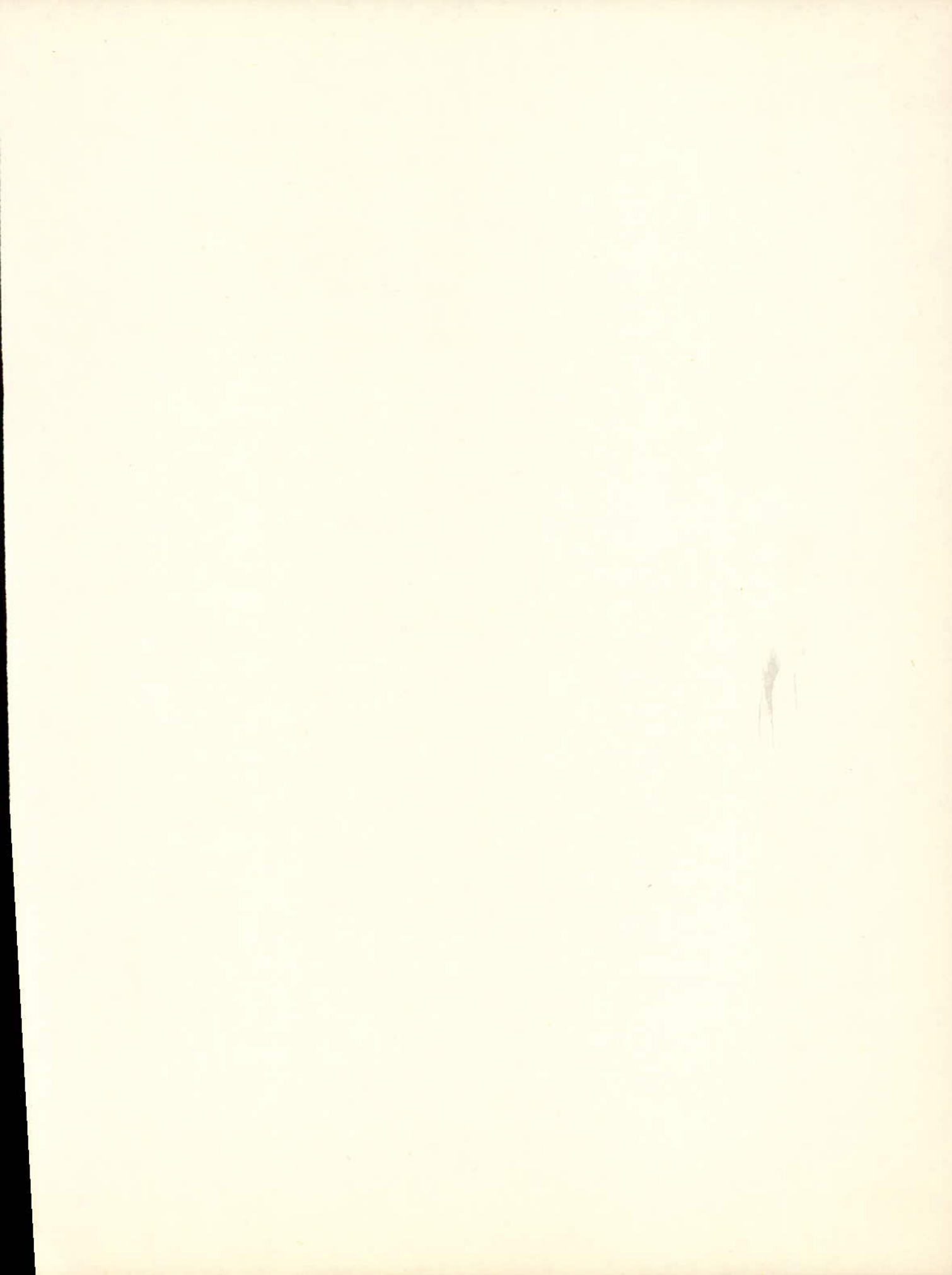
Avec ces formules on peut operer comme precedemment: l'inconvenient c'est que l'on doit calculer les derivees partielles de  $\phi$  ce qui peut etre complique, aussi on remplace ces derivees par des differences finies et on obtient les formules:

$$x_{(k+1)} = x_k + \phi(x_k, y_k) - \phi(x_k + h, y_k)$$

$$y_{(k+1)} = y_k + \phi(x_k, y_k) - \phi(x_k, y_k + h)$$

On peut partir de  $x_0 = y_0 = 0$  et  $h = 1$ . Si  $\phi(x_{k+1}, y_{k+1})$  est plus petit que  $\phi(x_k, y_k)$ , on continue, sinon on remplace  $h$  par  $h/2$ . On continue ainsi jusqu'a ce que  $h < \epsilon$  ou  $\epsilon$  est fixe a priori. Cette methode peut etre etendue au cas d'un nombre quelconque de variables. Elle converge toujours sauf dans quelques rares cas ou par suite de symetrie, on ne peut demarrer:

Par exemple avec  $x_0 = y_0 = 0$  et  $F(Z) = Z^2 + 1 = 0$



$$\begin{aligned}
& -f(i,j) \left( \frac{P(x,y)}{\Delta_{i+1}^i} + \frac{P(x,y)}{\Delta_i^i} + \frac{Q(x,y)}{\Delta_{j+1}^j} + \frac{Q(x,y)}{\Delta_j^j} \right) \\
& +f(i,j) \left( \frac{P(x,y)}{\Delta_{i+1}^i} + \frac{S(x,y)}{\Delta_{i+1}^i} \right) + f(i-1,j) \left( \frac{P(x,y)}{(\Delta_i^i)^2} - \frac{S(x,y)}{\Delta_{i+1}^i \Delta_i^i} \right) \\
& +f(i,j) \left( \frac{Q(x,y)}{\Delta_{j+1}^j} + \frac{T(x,y)}{\Delta_j^j} \right) + f(i,j-1) \left( \frac{Q(x,y)}{(\Delta_j^j)^2} - \frac{T(x,y)}{\Delta_j^j \Delta_{j+1}^j} \right) \\
& +2R(x,y) f(i+1,j+1) - 2R(x,y) f(i-1,j-1) \\
& \frac{(\Delta_i^i + \Delta_{i+1}^i)(\Delta_j^j + \Delta_{j+1}^j)}{(\Delta_i^i + \Delta_{i+1}^i)(\Delta_j^j + \Delta_{j+1}^j)} (\Delta_i^i + \Delta_{i+1}^i)(\Delta_j^j + \Delta_{j+1}^j) \\
& \frac{-2R(x,y) f(i-1,j+1) + 2R(x,y) f(i-1,j-1) + U(x,y)}{(\Delta_i^i + \Delta_{i+1}^i)(\Delta_j^j + \Delta_{j+1}^j)} = 0
\end{aligned}$$

On notera  $f(i,j) = f(x_i, y_j)$   $i=1, N$   $j=1, M$

en faisant varier  $i$  et  $j$  on obtient un système linéaire de la forme  $AX = B$  possédant  $M.N$  équations.

### III - REPLISSAGE DE A ET DE B

Le remplissage de la matrice  $A$  sera obtenu en parcourant la grille des points (maillage) de bas en haut et de gauche à droite.

En un point donné de la grille nous aurons une équation. La matrice  $A$  sera donc carrée et d'ordre  $M.N$ . Le remplissage du vecteur  $B$  sera obtenu en faisant parcourir la fonction  $U(x,y)$  aux différents points intérieurs de la grille et aux points extérieurs qui ne sont autres les conditions aux bords.

Le vecteur  $X$  aura pour composantes les valeurs de la fonction en chaque point de la grille.



## C - UNIFORMISATION DE L'ERREUR

Pour uniformiser l'erreur ( écart entre les vraies valeurs et les valeurs obtenues après la résolution par les méthodes itératives ), nous procéderons de la manière suivante: Nous divisons le maillage en k parties distinctes. Ceci nous conduit à décomposer la matrice d'iteration A en  $k^2$  blocs. NOUS calculons ensuite la norme vectorielle (norme du max) de chaque bloc. Nous formons une nouvelle matrice N( A ) dont les éléments seront les normes calculées précédemment. Nous calculons les valeurs propres de cette nouvelle matrice et nous les rapprochant en minimisant une fonction appropriée. Cette dernière opération a pour but de répartir uniformément l'erreur

### I- CAS GENERAL DE MAILLAGE IRREGULIER

#### I-1 DIVISION EN k BLOCS

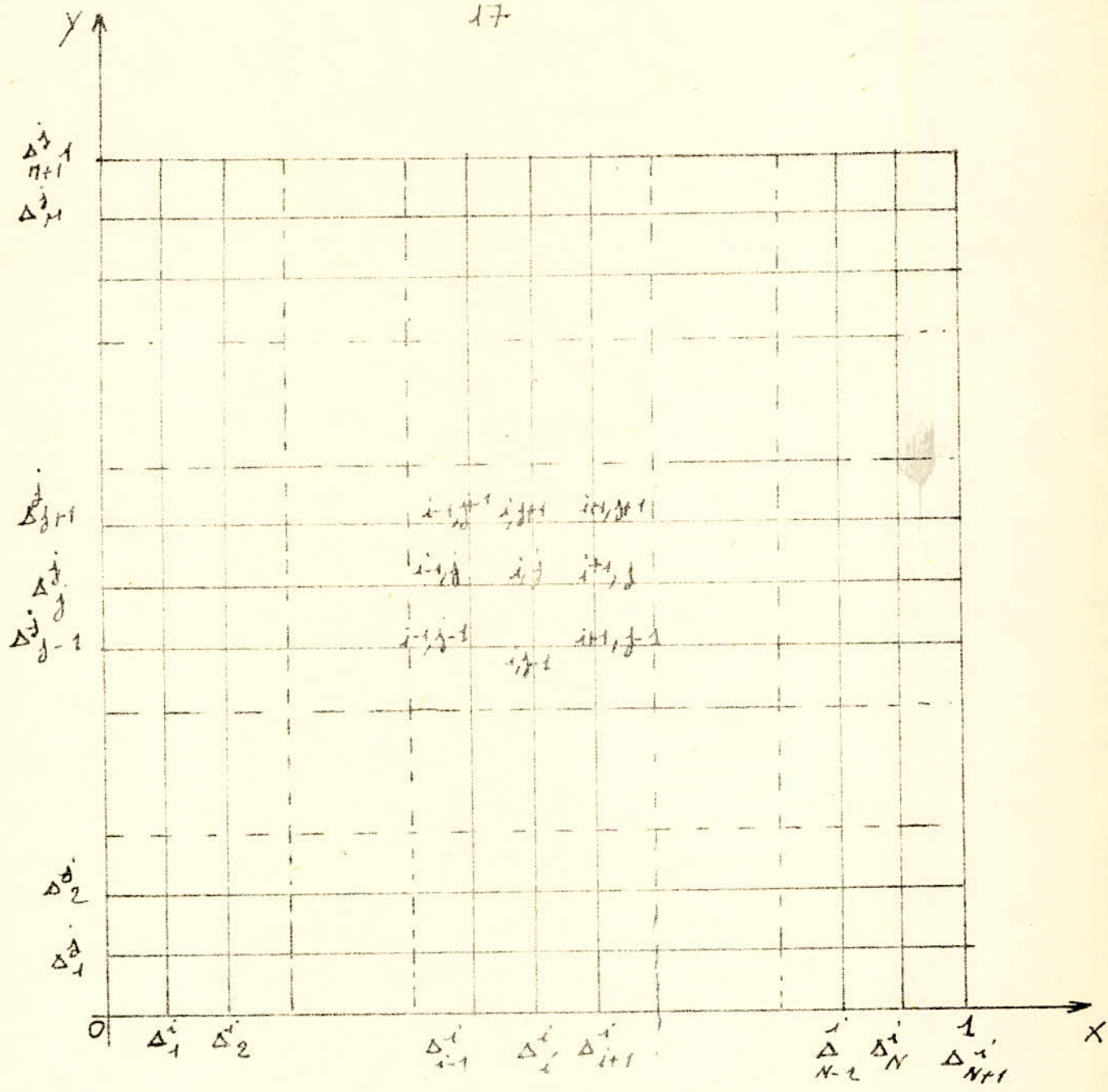
Le problème consiste à diviser les segments ( 0,1 ) suivant ox et oy en  $2k-1$  parties dont le milieu comporte  $n_1$  intervalles d'épaisseur  $D_1$ , de part et d'autre un deuxième maillage plus large d'épaisseur  $D_2$ . Au fur et à mesure que l'on s'écarte du milieu le maillage s'élargit. Nous obtenons de cette façon k maillages différents qui vont donner lieu à  $k^2$  blocs dans la matrice d'iteration de JACOBI ou de GAUSS-SEIDEL. La répartition des maillages doit vérifier la relation suivante:

$$nD_1 + 2n_2D_2 + 2n_3D_3 + \dots + 2n_kD_k = 1$$

Il suffit donc de fixer les k-1 premiers maillages pour obtenir le k-ième.

$$D_k = \frac{1}{2n_k} \left( 1 - n_1D_1 - 2 \sum_{i=2}^{k-1} n_iD_i \right)$$

NOUS supposons que nous avons la même répartition suivant ox et oy et que seul le dernier peut varier pour les deux.



I - 2 - CALCUL DES INDICES

Ces indices vont nous permettre de delimiter chaque partie du maillage suivant les variations de  $i$  et  $j$ . Ces derniers nous renseignent sur le nombre d'intervalles.

$$\begin{aligned}
 I_1 &= 0 \\
 I_2 &= I_1 + n_k \\
 I_3 &= I_2 + n_{k-1} \\
 &\vdots \\
 &\vdots \\
 I_{l+1} &= I_{l-1} + n_{k-l+2} \\
 &\vdots \\
 &\vdots \\
 I_{k+1} &= I_k + n_1
 \end{aligned}$$

Les differents domaines de variation de  $i$  et  $j$  sont les suivants:

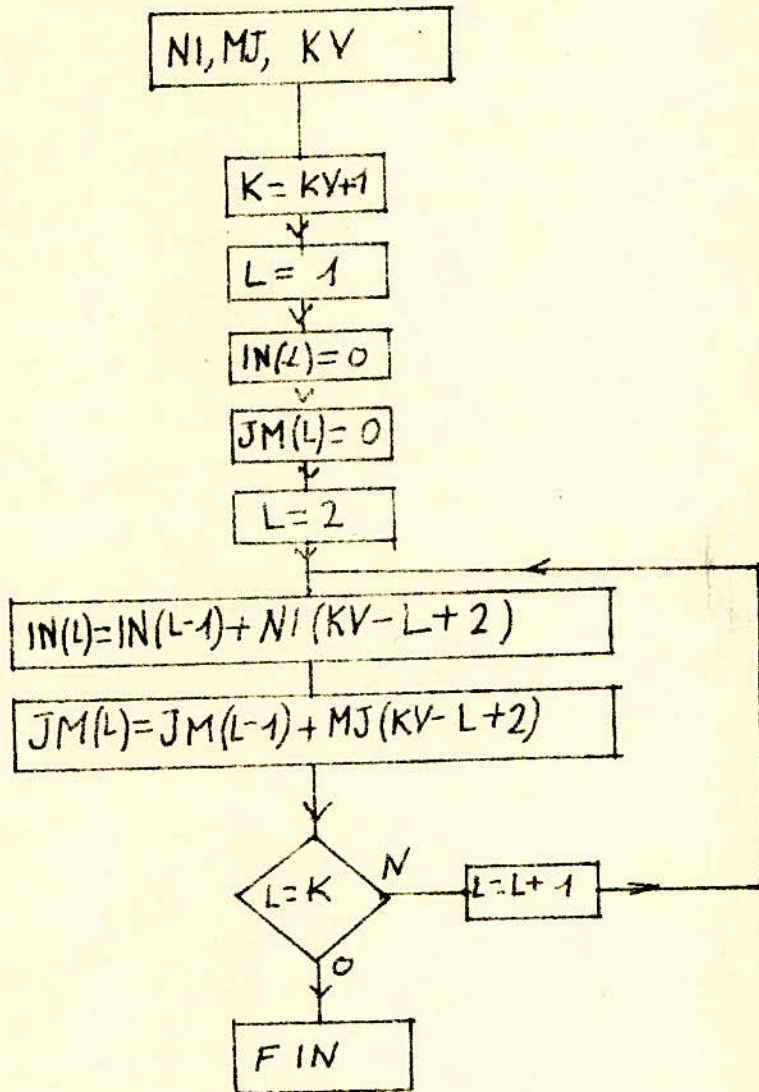
$$\begin{aligned}
 ( I_1 + 1, I_2 ) & \cup ( N+2 - I_2, N+2 - I_1 + 1 ) \\
 ( I_2 + 1, I_3 ) & \cup ( N+2 - I_3, N+2 - (I_2 + 1) ) \\
 & \vdots \\
 & \vdots \\
 ( I_{l+1} + 1, I_{l+1} ) & \cup ( N+2 - I_{l+1}, N+2 - (I_l + 1) ) \\
 & \vdots \\
 & \vdots \\
 ( I_{k+1} + 1, I_{k+2} ) & \cup ( N+2 - I_{k+2}, N+2 - (I_{k+1} + 1) )
 \end{aligned}$$

Il est de meme pour  $j$ .

$$( l = 2, k+1 )$$

Nous faisons varier  $i$  et  $j$  a partir des extremités  $( 1, N )$  et  $( 1, M )$ . Toutefois pour eviter toute confusion lors du passage du programme on procede a des changements de notations .Ainsi:

$$\begin{aligned}
 I &\rightarrow IN & J &\rightarrow JM \\
 n_i &\rightarrow NI(I) & n'_i &\rightarrow MJ(I) & k &\rightarrow k_v
 \end{aligned}$$

ORGANIGRAMME DU CALCUL DES INDICES

C SOUS PROGRAMME DU CALCUL DES INDICES

SUBROUTINE INDICE ( NI , MJ , KV , IN , JM )

DIMENSION NI ( 10 ) , MJ ( 10 ) , IN ( 10 ) , JM ( 10 )

K = KV + 1

L = 1

IN ( L ) = 0

JM ( L ) = 0

DO 1 L = 2 , k

IN ( L ) = IN ( L - 1 ) + NI ( KV - L + 2 )

JM ( L ) = JM ( L - 1 ) + MJ ( KV - L + 2 )

1 CONTINUE

RETURN

END

I - 3 - DISTRIBUTION DES MAILLAGES

Le but de cette distribution est de répartir les maillages suivant les variations de  $i$  et  $j$ . On rappelle qu'il n'est pas nécessaire de donner les  $k$  maillages puisque en fixant les  $k-1$  premiers on obtient le  $k$ ème.

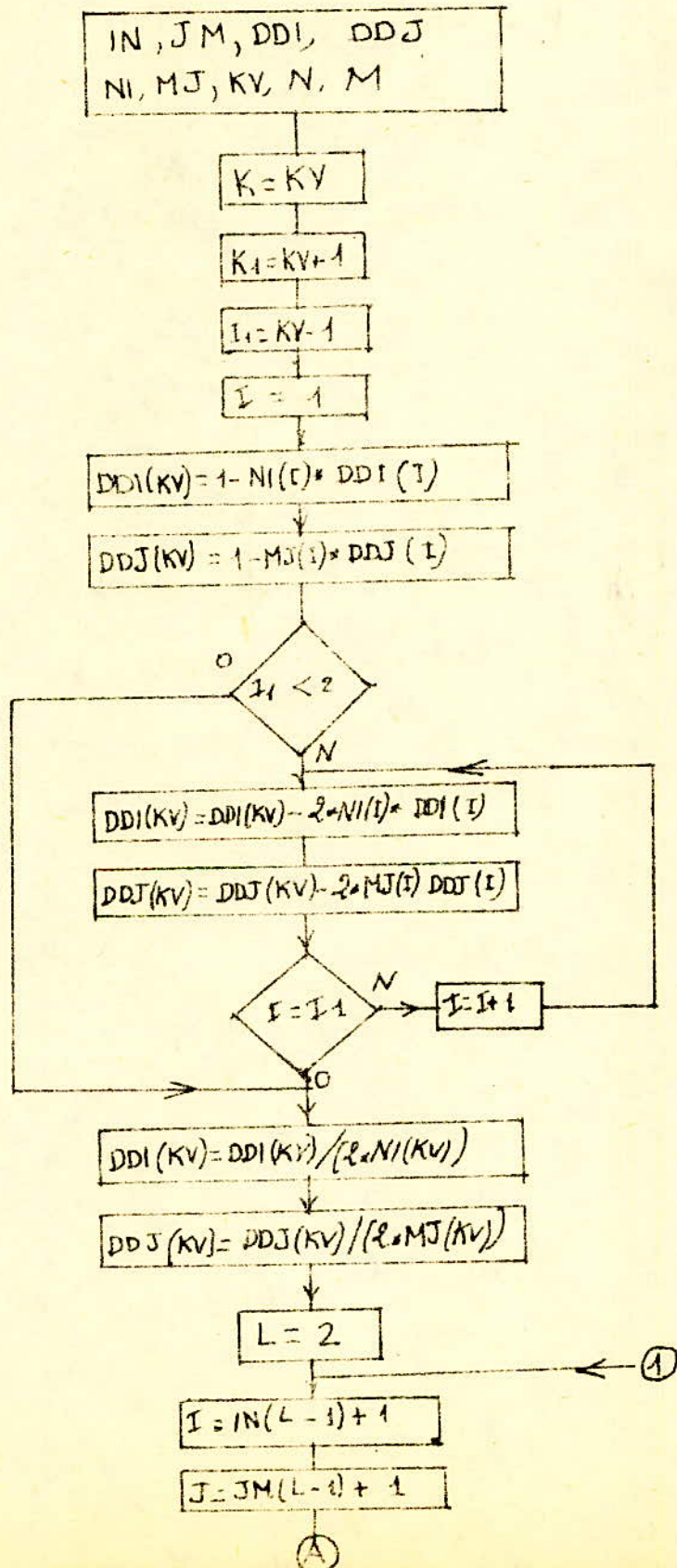
On prendra comme notations  $DDI(k_v)$  pour la distribution du maillage suivant  $ox$  et  $DDJ(k_v)$  suivant  $oy$ . On obtient l'algorithme suivant:

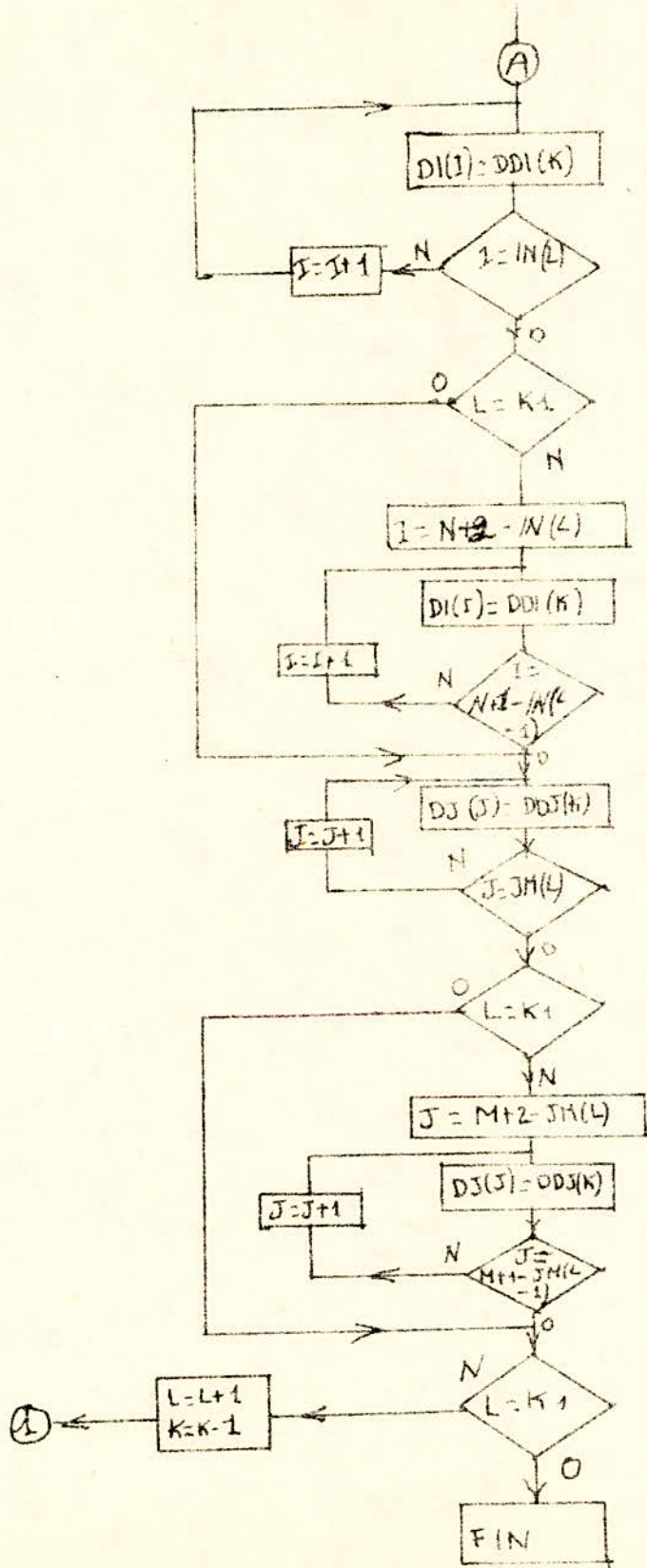
$$DDI(k_v) = \frac{(1 - NI(1) DDI(1) - 2 \sum_{I=2}^{k-1} NI(I) DDI(I))}{2NI(k_v)}$$

$$DDJ(k_v) = \frac{(1 - MJ(1) DDJ(1) - 2 \sum_{I=2}^{k-1} MJ(I) DDJ(I))}{2MJ(k_v)}$$

$DDI(k)$  et  $DDJ(k)$  représentent les épaisseurs du maillage respectivement suivant  $ox$  et  $oy$ .

# ORGANIGRAMME DE DISTRIBUTION DES MAILLAGES







C SOUS PROGRAMME DE DISTRIBUTION DE MAILLAGE

```

SOUBROUTINE DISTRI(IN, JM, DJ, DI, NI, MJ, KV, N, M, DDI, DDJ)
1 DIMENSION(IN(10), JM(10), DI(101), DJ(101), DDI(101), DDJ(101),
  NI(10), MJ(10))
  K = KV
  K1 = KV+1
  I1 = KV-1
  I = 1
  DDJ(I) = DDI(I)
  DDI(KV) = 1 - NI(I)*DDI(I)
  DDJ(KV) = 1 - MJ(I)*DDJ(I)
  IF(I1-2) 1,2,2
2 DO3 I=2, I1
  DDJ(I) = DDI(I)
  DDI(KV) = DDI(KV) - 2*NI(I)*DDI(I)
  DDJ(KV) = DDJ(KV) - 2*MJ(I)*DDJ(I)
3 CONTINUE
1 DDI(KV) = DDI(KV) / 2*NI(KV)
  DDJ(KV) = DDJ(KV) / 2*MJ(KV)
  DO5 L=2, K1
  I2 = IN(L-1) + 1
  J2 = JM(L-1) + 1
  I3 = IN(L)
  DO6 I=I2, I3
  DI(I) = DDI(K)
  IF(L-K1) 7,8,8
7 I2 = N+2 - I2
  I3 = N+2 - I3
  DO7 I=I3, I2
  DI(I) = DDI(K)
8 J2 = JM(L)
  DO10 J=J2, J3
10 DJ(J) = DDJ(K)
  IF(L-K1) 11,5,5
11 J2 = M+2 - J2
  J3 = M+2 - J3
  DO12 J=J3, J2
12 DJ(J) = DDJ(K)
  K = K-1
5 CONTINUE
  RETURN
  ENF

```

## II - DETERMINATION DE LA MATRICE D'ITERATION

Soient  $A_1$  et  $A_2$  respectivement les matrices de JACOBI et de GAUSS-SEIDEL associees a  $A$

Nous avons:

$$A = D - E - F$$

$$A = ( a_{ij} ), D = ( a_{ii} ), E = \begin{pmatrix} -a_{ij} \\ i > j \end{pmatrix}, F = \begin{pmatrix} -a_{ij} \\ i < j \end{pmatrix}$$

L'equation  $A X = B$  se transforme en :

$$( D - E - F ) X = B$$

Pour JACOBI on pose:

$$D X = B + ( E + F ) X \text{ d'ou on a:}$$

$$X^{(p+1)} = D^{-1} B + D^{-1} ( E + F ) X^{(p)}$$

$$\text{avec } A_1 = D^{-1} ( E + F ) \text{ et } B_1 = D^{-1} B$$

on aura :

$$X^{(p+1)} = A_1 X^{(p)} + B_1 \text{ ou encore}$$

$$x_i^{(p+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j^{(p)} \right), \quad A_1 = \begin{cases} 0 & \text{si } i = j \\ \frac{-a_{ij}}{a_{ii}} & \text{si } i \neq j \end{cases}$$

Pour GAUSS-SEIDEL on pose:

$$( D - E ) X = B + F X \text{ d'ou}$$

$$X^{(p+1)} = ( D - E )^{-1} B + ( D - E )^{-1} F X^{(p)}$$

$$\text{avec } A_2 = ( D - E )^{-1} F \text{ et } B_2 = ( D - E )^{-1} B$$

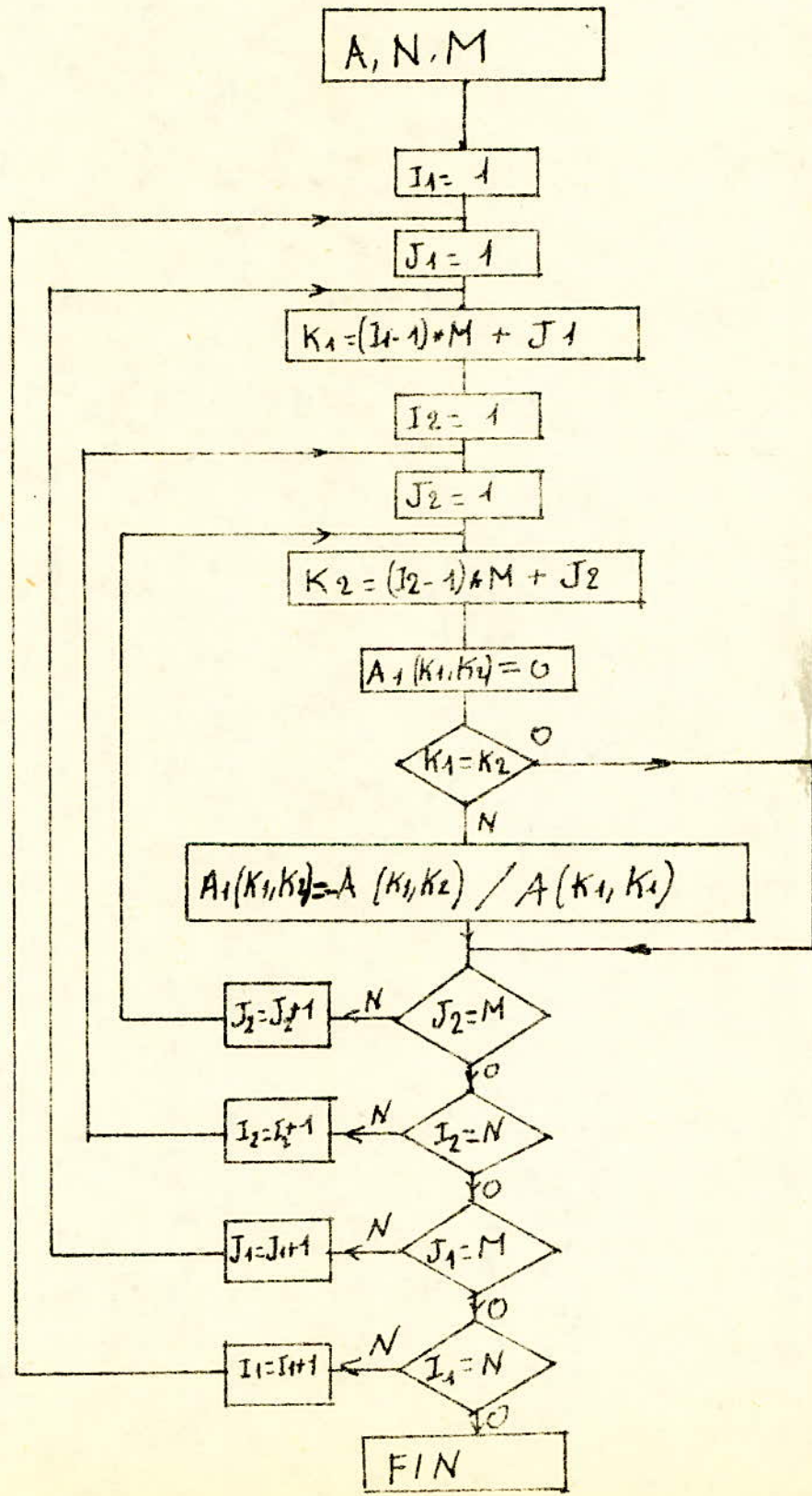
on aura :

$$X^{(p+1)} = A_2 X^{(p)} + B_2 \text{ ou encore:}$$

$$x_i^{(p+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(p+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(p)} \right)$$

Nous choisissons pour la suite de notre travail, la matrice  $A_1$  itérative de JACOBI associée à  $A$ . Cette dernière s'obtient plus facilement que la deuxième matrice de GAUSS-SEIDL qui exige un calcul compliqué des éléments d'une matrice inverse. D'autre part nous aurons réduit le nombre d'opération à effectuer et par la même occasion le temps d'exécution.

# ORGANIGRAMME DU CALCUL DE LA MATRICE D'ITERATION DE JACOBI



C SOUS PROGRAMME DE CALCUL DE A1

```
SOUBROUTINE ATILDA (A,N,M,A1)
DIMENSION (A(100,100),A1(100,100))
DO20 I1=1,N
DO20 J1=1,M
K1 = (I1 - 1)*M + J1
DO20 I2=1,N
DO20 J2=1,M
K2 = (I2 - 1)*M + J2
A1(K1,K2) = 0
IF(K1.NE.K2) A1(K1,K2) = -A(K1,K2) / A(K1,K1)
20 CONTINUE
RETURN
END
```

### III - CALCUL DES NORMES VECTORIELLES

Une fois la matrice d'iteration  $A_1$  determinee, nous calculons les differentes normes de chaque bloc. Nous rappelons que ces blocs sont engendres par les combinaisons possibles entre les indices I et J. Ainsi lorsque I et J varient dans les intervalles les plus exterieurs nous aurons la premiere norme, puisqu'il y a K domaines pour I et de meme pour J nous aurons donc  $K^2$  normes. La structure de la matrice d'iteration  $A_1$  sera la suivante:

$$A_1 = \begin{pmatrix} A_{11} & A_{12} & & A_{1k} \\ A_{21} & A_{22} & & A_{2k} \\ & & & \\ & & & \\ A_{k1} & A_{k2} & \dots & A_{kk} \end{pmatrix} \Rightarrow \begin{pmatrix} S_{11} & S_{12} & & S_{1k} \\ S_{21} & S_{22} & & S_{2k} \\ & & & \\ & & & \\ S_{k1} & S_{k2} & & S_{kk} \end{pmatrix} = S$$

S est une matrice carree dont les elements sont les normes vectorielles des blocs  $A_{IJ}$ . Pour determiner les  $S_{IJ}$  on peut soit:

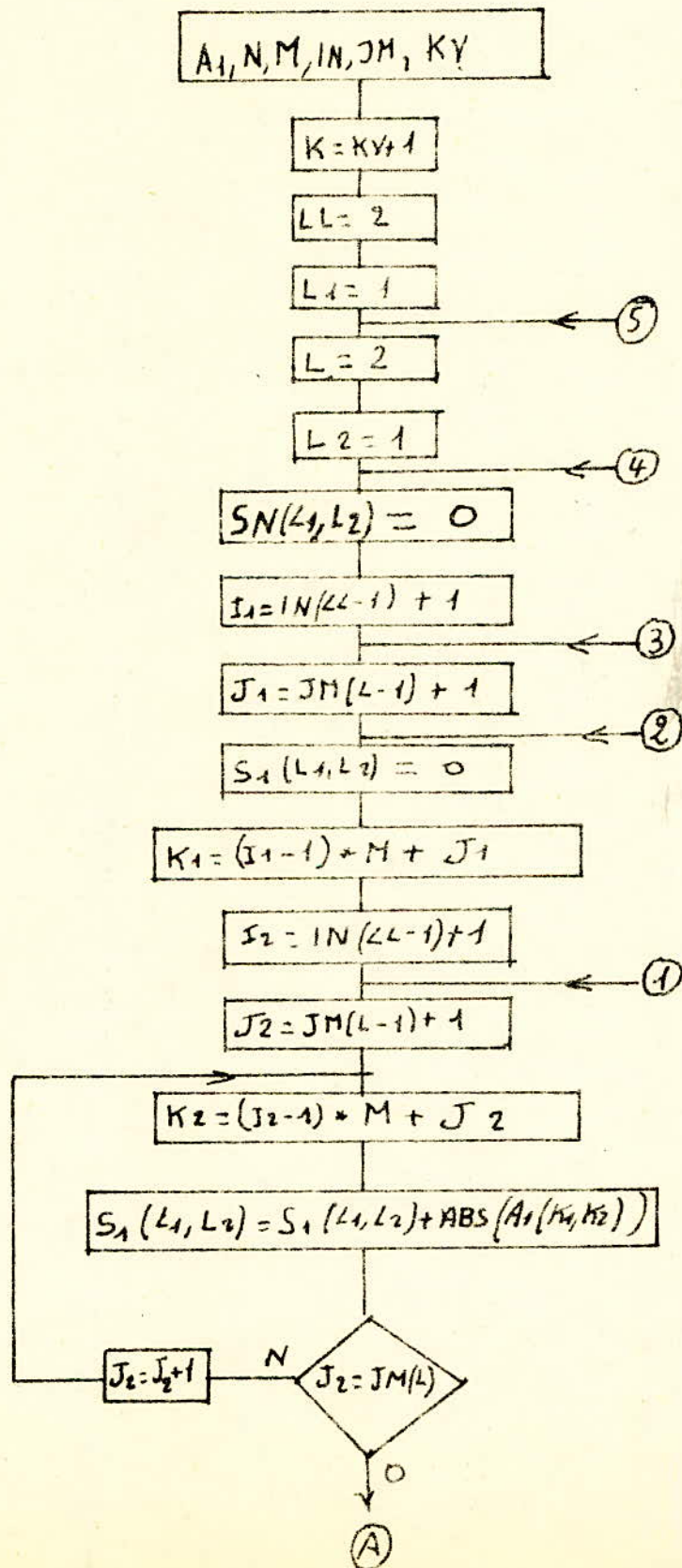
- calculer les elements de chaque bloc et ensuite sa norme.
- faire le calcul simultanement.

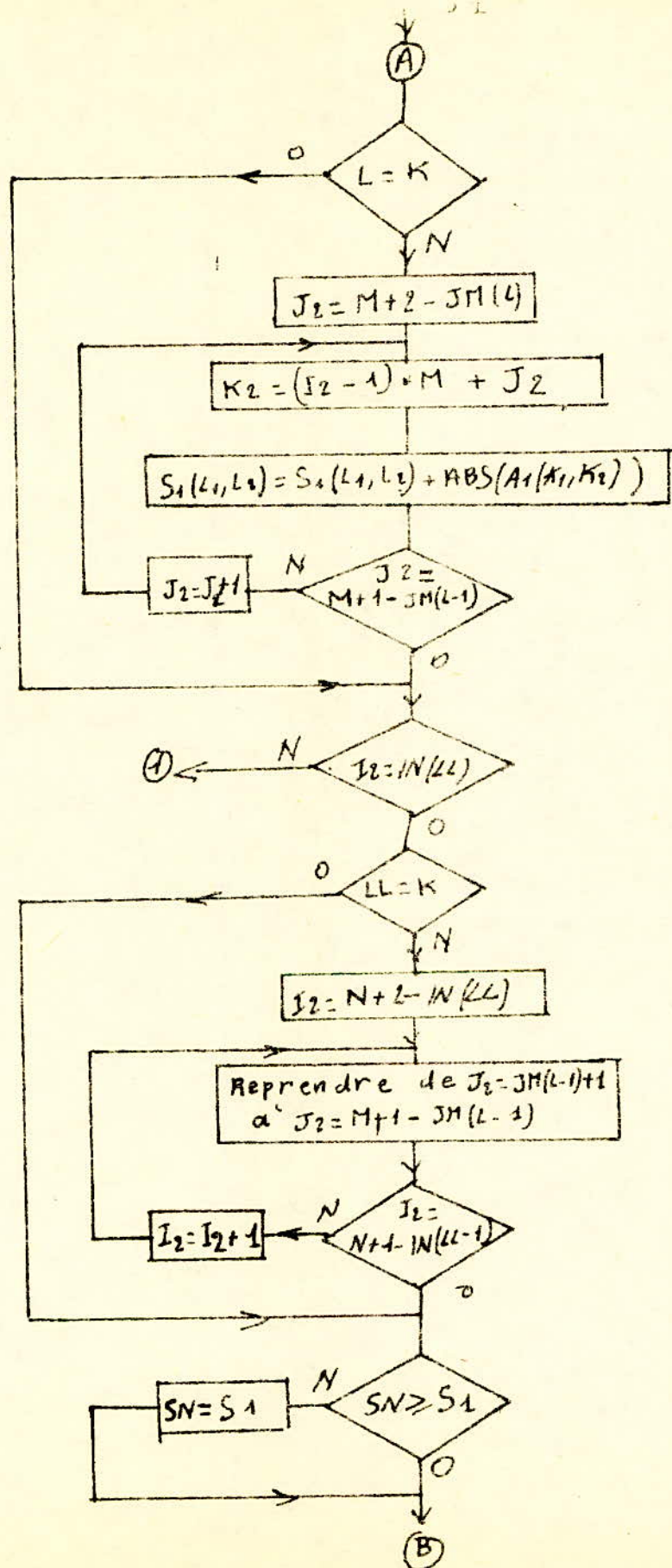
Le premier cas est simple. Le deuxieme est complique en raison de ses combinaisons d'indices, mais il presente l'avantage d'exiger moins d'operation a effectuer. Nous le choisirons ainsi que la norme du max.

On rappelle que la norme du max est la suivante:

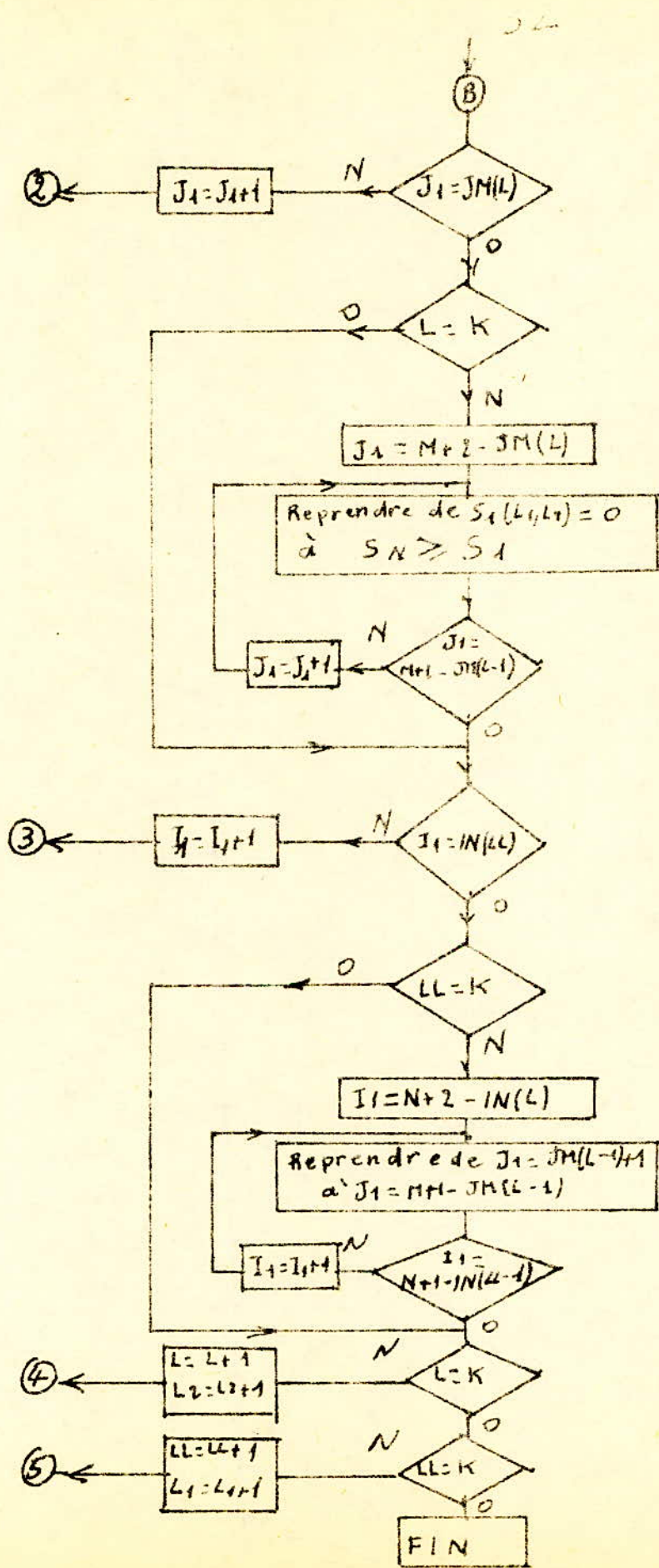
$$S(L_1, L_2) = \max_{\substack{K_1 \in D_1 \\ K_2 \in D_2}} \left( \sum_{i=1}^n A_1(K_1, K_2) \right)$$

# ORGANIGRAMME DU CALCUL DES NORMES VECTORIELLES









C - SOUS PROGRAMME DU CALCUL DES NORMES VECTORIELLES

SUBROUTINE NORMES (A1,N,M,IN,JM,KV,SN,S1)

DIMENSION A1(100,100), IN(10), JM(10), SN(10,10), S1(10,10)

K= KV+1

L1= 0

DO3OLL= 2,K

L1= L1+1

L2= 0

DO3OL= 2,K

L2= L2+1

SN(L1,L2)= 0

I5= IN(LL-1)+1

I6= IN(LL)

DO37I1= I5,I6

J5= JM(L-1)+1

J6= JM(L)

DO38J1= J5,J6

S1(L1,L2)= 0

K1= (I1-1)\*M+J1

I3= IN(LL-1)+1

I4= IN(LL)

DO39I2= I3,I4

J3= JM(L-1)+1

J4= JM(L)

DO40J2= J3,J4

K2= (I2-1)\*M+J2

S1(L1,L2)=S1(L1,L2)+ABS(A1(K1,K2))

40 CONTINUE

IF(L-K)5,39,39

5 J4= M+2-J4

J3= M+2-J3

DO39J2= J4,J3

K2= (I2-1)\*M+J2

S1(L1,L2)= S1(L1,L2)+ABS(A1(K1,K2))

```
39 CONTINUE
   IF(LL-K)6,7,7
   6 I4= N+2-I4
     I3= N+2-I3
     DO42I2= I4,I3
     J3= JM(L-1)+1
     J4= JM(L)
     DO43J2= J3,J4
     K2= (I2-1)*M+J2
     S1(L1,L2)= S1(L1,L2)+ABS(A1(K1,K2))
43 CONTINUE
   IF(L-K)8,42,42
   8 J4= M+2-J4
     J3= M+2-J3
     DO42J2= J4,J3
     K2= (I2-1)*M+J2
     S1(L1,L2)= S1(L1,L2)+ABS(A1(K1,K2))
42 CONTINUE
   7 IF(SN(L1,L2)-S1(L1,L2))58,38,38
58 SN(L1,L2)=S1(L1,L2)
38 CONTINUE
   IF(L-K)9,37,37
   9 J5= M+2-J5
     J6= M+2-J6
     DO37J1= J6,J5
     S1(L1,L2)= 0
     K1= (I1-1)*M+J1
     I3= IN(LL-1)+1
     I4= IN(LL)
     DO45I2= I3,I4
     J3= JM(L-1)+1
     J4= JM(L)
     DO46J2= J3,J4
     K2= (I2-1)*M+J2
     S1(L1,L2)= S1(L1,L2)+ABS(A1(K1,K2))
```

```

46 CONTINUE
   IF(L-K)11,45,45
11 J4= M+2-J4
   J3= M+2-J3
   DO45J2= J4,J3
   K2= (I2-1)*M+J2
   S1(L1,L2)= S1(L1,L2)+ABS(A1(K1,K2))
45 CONTINUE
   IF(LL-K)12,71,71
12 I4= N+2-I4
   I3= N+2-I3
   DO47I2= I4,I3
   J3= JM(L-1)+1
   J4= JM(L)
   DO48J2= J3,J4
   K2= (I2-1)*M+J2
   S1(L1,L2)= S1(L1,L2)+ABS(A1(K1,K2))
48 CONTINUE
   IF(L-K)13,47,47
13 J4=M+2-J4
   J3= M+2-J3
   DO47J2= J4,J3
   K2= (I2-1)*M+J2
   S1(L1,L2)= S1(L1,L2)+ABS(A1(K1,K2))
47 CONTINUE
71 IF(SN(L1,L2)-S1(L1,L2))59,37,37
59 SN(L1,L2)= S1(L1,L2)
37 CONTINUE
   IF(LL-K)14,30,30
14 I5= N+2-I5
   I6= N+2-I6
   DO30I1= I6,I5
   J5= JM(L-1)+1
   J6= JM(L)
   DO49J1= J5,J6
   S1(L1,L2)= 0

```

```

K1= (I1-1)*M+J1
I3= IN(LL-1)+1
I4= IN(LL)
DO50I2= I3,I4
J3= JM(L-1)+1
J4= JM(L)
DO51J2= J3,J4
K2= (I2-1)*M+J2
S1(L1,L2)= S1(L1,L2)+ABS(A1(K1,K2))
51 CONTINUE
IF(L-K)15,50,50
15 J4= M+2-J4
J3= M+2-J3
DO50J2= J4,J3
K2= (I2-1)*M+J2
S1(L1,L2)= S1(L1,L2)+ABS(A1(K1,K2))
50 CONTINUE
IF(LL-K)16,17,17
16 I4= N+2-I4
I3= N+2-I3
DO52I2= I4,I3
J3= JM(L-1)+1
J4= JM(L)
DO53J2= J3,J4
S1(L1,L2)= S1(L1,L2)+ABS(A1(K1,K2))
53 CONTINUE
IF(L-K)18,52,52
18 J4= M+2-J4
J3= M+2-J3
DO52J2= J4,J3
K2= (I2-1)*M+J2
S1(L1,L2)= S1(L1,L2)+ABS(A1(K1,K2))
52 CONTINUE
17 IF(SN(L1,L2)-S1(L1,L2))60,49,49
60 SN(L1,L2)= S1(L1,L2)

```

```

49 CONTINUE
   IF(L-K)19,30,30
19 J5= M+2-J5
   J6= M+2-J6
   DO30J1= J6,J5
   S1(L1,L2)= 0
   K1= (I1-1)*M+J1
   I3= IN(LL-1)+1
   I4= IN(LL)
   DO54I2= I3,I4
   J3= IN(L-1)+1
   J4= JM(L)
   DO55J2= J3,J4
   K2= (I2-1)*M+J2
   S1(L1,L2)= S1(L1,L2)+ABS(A1(K1,K2))
55 CONTINUE
   IF(L-K)20,54,54
20 J4= M+2-J4
   J3= M+2-J3
   DO54J2= J4,J3
   K2= (I2-1)*M+J2
   S1(L1,L2)= S1(L1,L2)+ABS(A1(K1,K2))
54 CONTINUE
   IF(LL-K)21,72,72
21 I4= N+2-I4
   I3= N+2-I3
   DO73I2= I4,I3
   J3= JM(L-1)+1
   J4= JM(L)
   DO56J2= J3,J4
   K2= (I2-1)*M+J2
   S1(L1,L2)= S1(L1,L2)+ABS(A1(K1,K2))
56 CONTINUE
   IF(L-K)22,73,73
22 J4= M+2-J4
   J3= M+2-J3

```

```
DO73J2= J4,J3
K2= (I2-1)*M+J2
S1(L1,L2)= S1(L1,L2)+ABS(A1(K1,K2))
73 CONTINUE
72 IF(SN(L1,L2)-S1(L1,L2))61,30,30
61 SN(L1,L2)= S1(L1,L2)
30 CONTINUE
RETURN
END
```

IV - CALCUL DES VALEURS ET VECTEURS PROPRES

Après avoir déterminé la matrice SN des normes, nous calculons ses valeurs propres et ses vecteurs propres.

Son polynôme caractéristique est le suivant:

$$P(\lambda) = \det(SN(I, J) - \lambda I) = \begin{vmatrix} SN(1,1) - \lambda & SN(1,2) & \dots & SN(1, KV) \\ SN(2,1) & SN(2,2) - \lambda & \dots & SN(2, KV) \\ \dots & \dots & \dots & \dots \\ SN(KV,1) & \dots & \dots & SN(KV, KV) - \lambda \end{vmatrix}$$

Pour ce faire nous utilisons le théorème de la puissance itérée en supposant que toutes les valeurs propres sont distinctes et simples.

Dans le cas où la matrice SN est à deux dimensions (2x2),

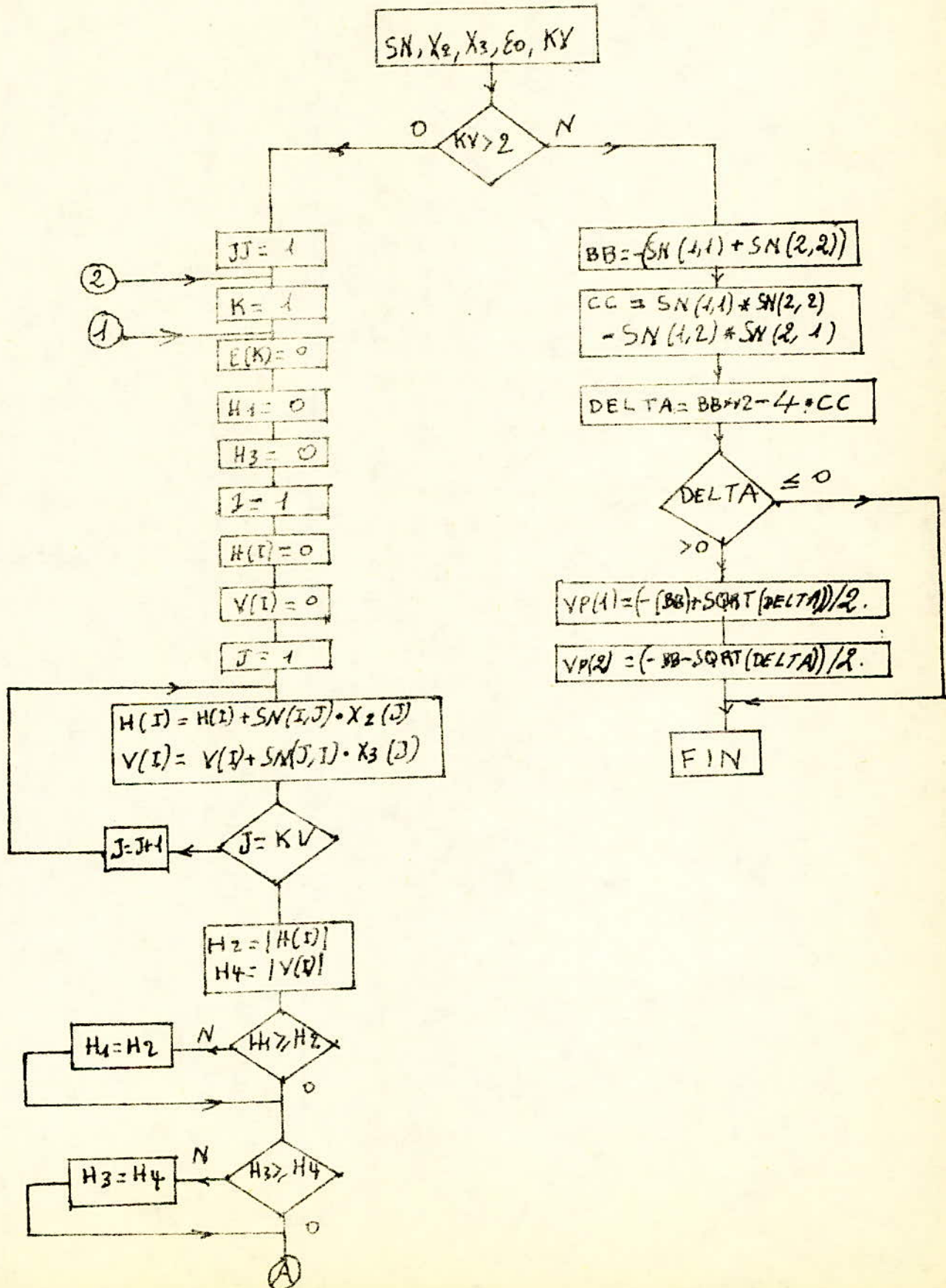
C'est la division en quatre blocs, il est préférable de résoudre une équation du second degré en  $\lambda$  pour trouver  $\lambda_1$  et  $\lambda_2$  en effet,

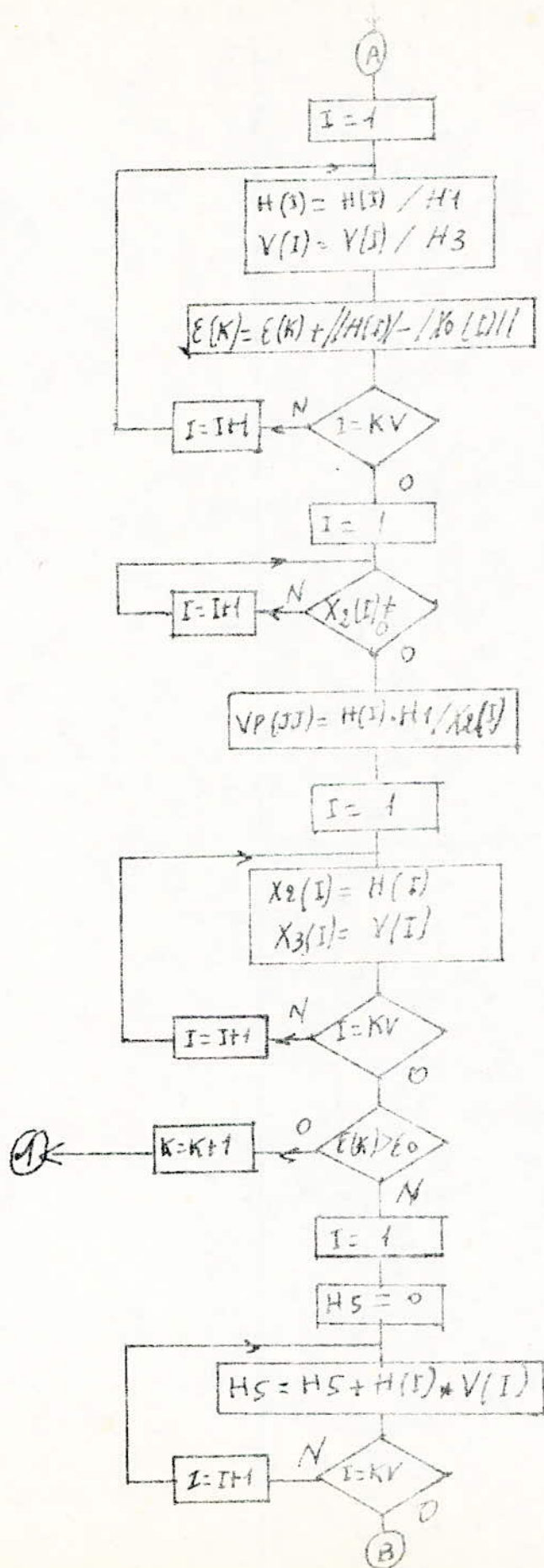
$$P(\lambda) = \det \begin{vmatrix} SN(1,1) - \lambda & SN(1,2) \\ SN(2,1) & SN(2,2) - \lambda \end{vmatrix} = 0$$

$$P(\lambda) = \lambda^2 - \lambda(SN(1,1) + SN(2,2)) + SN(1,1)SN(2,2) - SN(1,2)SN(2,1) = 0$$

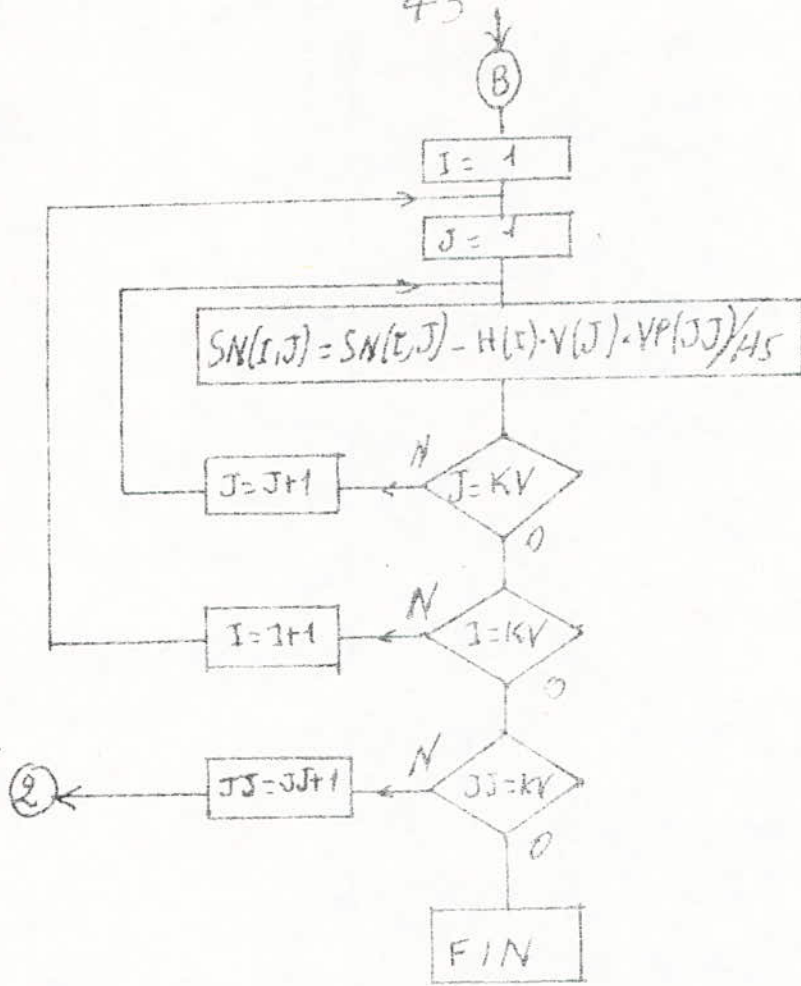


# ORGANIGRAMME DU CALCUL DES VALEURS PROPRES





43



C SOUS PROGRAMME DU CALCUL DES VALEURS PROPRES

```

SUBROUTINE VAL-PRO ( SN,X2,EPS2,KV,H,VP,V,X3 )
DIMENSION SN(10,10),X2(10),H(10),VP(10),EPS(100),V(10),X3(10)
IF(KV-2) 18,10,11
11 DO 1 JJ=1,KV
    K=1
8   EPS(K) = 0
    H1 = 0
    H3 = 0
    DO2I=1,KV
    H(I)=0
    V(I)=0
    DO3J=1,KV
    H(I)=H(I) +SN(I,J)*X2(J)
    V(I)=V(I)+SN(J,I)*X3(J)
3   CONTINUE
    H2=ABS(H(I))
    H4=ABS(V(I))
    IF(H1-H2)12,2,2
12  H1=H2
    IF(H3-H4)16,2,2
16  H3=H4
2   CONTINUE
    DO13I=1,KV
    H(I)=H(I)/H1
    V(I)=V(I)/H3
    EPS(K)=EPS(K) +ABS(ABS(H(I))-ABS(X2(I)))
13  CONTINUE
    I=0
4   I=I+1
    IF(X2(I)-0)14,15,14
15  GOTO4
14  VP(JJ)=H(I)*H1/X2(I)

```

-SUITE-

```

DO5I=1,KV
X(I)=H(I)
X3(I)=V(I)
5  CONTINUE
   IF(S(K)-S2)6,6,7
7  K=K+1
   GOTO8
6  H5=0
   DO17I=1,KV
   H5=H5+H(I)*V(I)
17 CONTINUE
   DO9I=1,KV
   DO9J=1,KV
   SN(I,J)=SN(I,J)-H(I)*V(J)*V(J)/H5
9  CONTINUE
1  CONTINUE
   GOTO18
10 BB=-(SN(1,1)+SN(2,2))
   CC=SN(1,1)*SN(2,2)-SN(1,2)*SN(2,1)
   DELTA=BB**2-4*CC
   IF(DELTA)18,18,20
20 VP(1)=(-BB+SORT(DELTA))/2.
   VP(2)=(-BB-SORT(DELTA))/2.
18 RETURN
   END

```

V- MINIMISATION

La minimisation consiste dans une première étape à réduire les écarts entre les différentes valeurs propres successives.

Dans une deuxième étape à empêcher ces valeurs propres d'augmenter rapidement afin de satisfaire le critère de convergence. La fonction appropriée sera la suivante:

$$F(D) = \sum_{i=1}^{kv-1} (|\lambda_{i+1}| - |\lambda_i|)^2 + \sum_{i=1}^{kv} R \lambda_i^2$$

Le premier terme  $(|\lambda_{i+1}| - |\lambda_i|)^2$  se vira à réduire l'écart et le deuxième terme  $R \lambda_i^2$  à satisfaire le critère de convergence. R est un paramètre que nous choisissons pour réaliser ce compromis.

Le but de cette minimisation est de répartir l'erreur uniformément à travers les maillages.

La variable D est un vecteur dont les composantes sont les kv-1 maillages qu'il faut trouver.

$$D = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ \vdots \\ D_{kv-1} \end{bmatrix} \quad \lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \vdots \\ \lambda_{kv} \end{bmatrix}$$

$$\lambda_i = f(D) \quad i=1, kv$$

POUR minimiser la fonction F(D), nous utilisons la méthode du gradient ou de plus grande pente, décrite dans le chapitre A. Ce qui nous donne:

$$\begin{aligned}
 D_1^{k+1} &= D_2^k + F(D_1^k, D_2^k, \dots, D_{kv-1}^k) - F(D_1^k + h, D_2^k, \dots, D_{kv-1}^k) \\
 D_2^{k+1} &= D_2^k + F(D_1^k, D_2^k, \dots, D_{kv-1}^k) - F(D_1^k, D_2^k + h, \dots, D_{kv-1}^k) \\
 &\vdots \\
 D_{kv-1}^{k+1} &= D_{kv-1}^k + F(D_1^k, D_2^k, \dots, D_{kv-1}^k) - F(D_1^k, D_2^k, \dots, D_{kv-1}^k + h)
 \end{aligned}$$

On se fixe un  $h$  de départ. Si  $F(D_1^{k+1}, D_2^{k+1}, \dots, D_{kv-1}^{k+1})$  est plus petit que  $F(D_1^k, D_2^k, \dots, D_{kv-1}^k)$  on continue

Sinon on remplace  $h$  par  $h/2$ . On continue ainsi le processus jusqu'à  $h < \varepsilon$  ou  $\varepsilon$  est la précision voulue.

C PROGRAMME PRINCIPAL

```

DIMENSION A(100,100),B(100),DI(101),DJ(101),F1(102),F2(102),
1 F3(100),F4(100),B1(100),X0(100),X(100),r1(100,100),B2(100),Z(100),
2 A2(100,100),C(10000),F(10,10),B3(100),NI(10),MJ(10),IN(10),JM(10),
3 DDI(10),SN(10,10),X2(10),H(10),VP(10),D2(10),D1(10),V(10),
4 X3(10),DDJ(10)
  READ(5,100)N,M
100  FORMAT(8I3)
  WRITE(6,140)N,M
140  FORMAT(1X,'PROGRAMMATION DES EQUATIONS A DERIVEES PARTIELLES',///,
  1X,'N='I3,'M='I3)
  NP1=N+1
  MP1=M+1
  READ(5,113)KV
113  FORMAT(8I3)
  K=KV-1
  READ(5,110)(DDI(I),I=1,K)
  WRITE(6,150)(DDI(I),I=1,K)
110  FORMAT(F6.3)
150  FORMAT(1X,'VECTEUR DDI:',//,2(1X,F6.3))
  READ(5,111)(NI(I),I=1,KV),(MJ(I),I=1,KV)
111  FORMAT(2I3)
  WRITE(6,112)(NI(I),I=1,KV),(MJ(I),I=1,KV)
112  FORMAT(2(1X,I3))
  READ(5,114)EPS2
114  FORMAT(E8.2)
  WRITE(6,117)EPS2
117  FORMAT(1X,'EPS2=',E8.2)
  READ(5,115)EPS3
115  FORMAT(E8.2)
  WRITE(6,120)EPS3
120  FORMAT(1X,'EPS3=',E8.2)
  READ(5,123)RR
123  FORMAT(E8.2)

```



-SUITE-

```

WRITE(6,124)RR
124  FORMAT(1X,'RR=',E8.2)
      CALL INDICE(NI,MJ,KV,IN,JM)
      READ(5,119)H3
119  FORMAT(E8.2)
      K=KV+1
      WRITE(6,116)((IN(I),I=1,K),(JM(I),I=1,K))
116  FORMAT(3(1X,I3))
      90  CALL DISTRI(IN,JM,DJ,DI,NI,MJ,KV,N,M,DDI,DDJ)
      WRITE(6,40)(DI(I),I=1,NP1),(DJ(J),J=1,MP1)
      40  FORMAT(1X,'VECTEUR DI ET DJ:',2(1X,F6.3))
      CALL REMA(N,M,DI,DJ,A)
      CALL ATILDA(A,N,M,n1)
      CALL NORMES(A1,N,M,IN,JM,KV,SN,S1)
      WRITE(6,41)((SN(I,J),J=1,KV),I=1,KV)
      41  FORMAT(1X,'NORMES VECTORIELLES:',//,2(1X,E13.6))
      I=1
      X(I)=1
      X3(I)=1
      DO99I=2,KV
      X2(I)=0
      X3(I)=0
      99  CONTINUE
      CALL VALPRO(SN,X2,EPS2,KV,H,VP,V,X3)
      WRITE(6,42)(VP(I),I=1,KV)
      42  FORMAT(1X,'VALEURS PROPRES:',//,2(1X,E13.6))
      L3=KV-1
      F5=0
      DO106J=1,L3
106  F5=F5+(ABS(VP(J+1))-ABS(VI(J)))*2+RR*VI(J)**2
      3  DO102K=1,L3
      D1(K)=DDI(K)
      DDI(K)=D1(K)+H3
      CALL DISTRI(IN,JM,DJ,DI,NI,MJ,KV,N,M,DDI,DDJ)

```

-SUITE-

```

CALL RLMA(N,M,DI,DJ,A)
CALL ATILDA(A,N,M,A1)
CALL NORMES(A1,N,M,IN,JM,KV,SN,S1)
CALL VALPRO(SN,X2,EPS2,KV,H,VP,V,X3)
F6=0
DO101J=1,L3
101 F6=F6+(ABS(VP(J+1))-ABS(VP(J)))**2+RF*VP(J)**2
   DDI(K)=D1(K)+F5-F6
   D2(K)=DDI(K)
   DDI(K)=D1(K)
102 CONTINUE
   DO103K=1,L3
103 DDI(K)=D2(K)
   CALL DISTRI(IN,JM,DJ,DI,NI,MJ,KV,N,M,DDI,DDJ)
   CALL RLMA(N,M,DI,DJ,A)
   CALL ATILDA(A,N,M,A1)
   CALL NORMES(A1,N,M,IN,JM,KV,SN,S1)
   CALL VALPRO(SN,X2,EPS2,KV,H,VP,V,X3)
   F6=0
   DO104J=1,L3
104 F6=F6+(ABS(VP(J+1))-ABS(VP(J)))**2+RR*VP(J)**2
   IF(F5-F6)1,2,2
     1 H3=H3/2
     WRITE(6,121)H3
121 FORMAT(1X,'H3 AVEC DIVISION=',E8.2)
     DO105K=1,L3
105 DDI(K)=D1(K)
     GOTO3
     2 IF(EPS3-H3)4,5,5
     4 GOTO90
     5 CALL DISTRI(IN,JM,DJ,DI,NI,MJ,KV,N,M,DDI,DDJ)
     WRITE(6,122)H3
122 FORMAT(1X,'H3 SANS DIVISION=',E8.2)
     WRITE(6,91)(DDI(I),I=1,KV)

```

-SUITE-

```

91  FORMAT(1X,'MEILLEUR MAILLAGE:',//,2(1X,E13.6))
    WRITE(6,118)(V(I),I=1,KV)
118  FORMAT(1X,'VALEURS PROPRES:',//,2(1X,E13.6))
    L=N+2
    Y=0
    DO500I=1,L
    F1(I)=FF1(Y)
    F2(I)=FF2(Y)
    Y=Y+DJ(I)
500  CONTINUE
    XX=DI(1)
    DO510I=1,N
    F3(I)=FF3(XX)
    F4(I)=FF4(XX)
    XX=XX+DI(I+1)
510  CONTINUE
    ZZ=0
    DO50I=1,N
    ZZ=ZZ+DI(I)
    Y=0
    DO50J=1,M
    Y=Y+DJ(J)
    F(I,J)=FF(ZZ,Y)
50  CONTINUE

C   -REPLISSAGE DU VECTEUR B-

    CALL REMB(N,M,DI,DJ,B,F1,F2,F3,F4)
    NN=N*M
    WRITE(6,170)(B(I),I=1,NN)
170  FORMAT(1X,'VECTEUR B:',///,4(1X,F6.2))

```

-SUITE-

```

C   REMPLISSAGE DE LA MATRICE A
      CALL REMA(N,M,DI,DJ,A)
      WRITE(6,180)
180  FORMAT(1X,'MATRICE A:',//)
      DO520I=1,NN
      WRITE(6,340)(A(I,J),J=1,NN)
340  FORMAT(1X,10E13.6)
520  CONTINUE
      DO550I=1,NN
      B2(I)=B(I)
      DO550J=1,NN
      A2(I,J)=A(I,J)
550  continue
      READ(5,220)EPS
220  FORMAT(E8.2)
      WRITE(6,360)EPS
360  FORMAT(1X,'EPS=',E8.2)
      K=1
      DO80I=1,N
      DO80J=1,M
      Z(K)=F(I,J)
      K=K+1
80  CONTINUE
      WRITE(6,76)(F(I,J),J=1,M),I=1,N)
76  FORMAT(1X,'VALEURS REELLES DE LA FONCTION:',//7(1X,E13.6))
      CALL RANGE(A,NN,C)
      NM=NN*NN

```

-SUITL-

```

C   RESOLUTION PAR METHODE DE GAUSS
      CALL GELG(B,C,NN,1,EPS,IER)
      K=1
      DO77I=1,NN
      B3(K)=B(I)
      K=K+1
77  CONTINUE
      WRITE(6,370)(B(I),I=1,NN)
370  FORMAT(1X,'SOLUTION PAR METHODE DE GAUSS: ',//,7(1X,E13.6))
      IF(IER.EQ.-1)WRITE(6,380)
380  FORMAT(1X,'CAS DE SOLUTION')
      READ(5,200)EPSO
200  FORMAT(E10.4)
      WRITE(6,410)EPSO
410  FORMAT(1X,'EPSO=',E10.4)
C   RESOLUTION PAR LA METHODE DE JACOBI
      CALL JACOBI(N,M,A2,B2,B,EPSO,X)
      WRITE(6,460)(X(I),I=1,NN)
460  FORMAT(1X,'SOLUTION PAR METHODE DE JACOBI: ',//,7(1X,E13.6))
      DO70K=1,NN
70   B3(K)=B3(K)-Z(K)
      WRITE(6,71)(B3(K),K=1,NN)
71   FORMAT(1X,'ERREUR PAR GAUSS: ',//,7(1X,E13.6))
      DO73K=1,NN
73   X(K) =X(K)-Z(K)
      WRITE(6,75)(X(K),K=1,NN)
75   FORMAT(1X,'ERREUR PAR JACOBI: ',//,7(1X,E13.6))
      CALL EXIT
      STOP
      END

```

D - RESOLUTION ET COMPARAISON

APRES avoir obtenu le meilleur maillage qui uniformise l'erreur on resoud l'equation  $AX=B$

Application au cas de quatre blocs

Nous avons  $k=2$   $n_1 = 4$   $n_2 = 2$   $n'_1 = 2$   $n'_2 = 3$   $R = 0$

$$H_3 = 10^{-3} \quad \Delta S_2 = 10^{-6} \quad \Delta S_3 = 510^{-6}$$

$N = M = 7$  apres minimisation on trouve:

$$\text{Suivant } x \quad D_1 = 0,123 \quad D_2 = 0,127$$

$$\text{Suivant } y \quad D_1 = 0,123 \quad D_2 = 0,126$$

L'equation aux derivees partielles est la suivante:

$$-\cos(x+y) \frac{\partial^2 f(x,y)}{\partial x^2} - \sin(x+y) \frac{\partial f(x,y)}{\partial y} = 1$$

$$P = -\cos(x+y)$$

$$Q = 0$$

$$R = 0$$

$$S = 0$$

$$T = -\sin(x+y)$$

$$U = -1$$

Nous vous donnons plus loin les rsultats trouves par la methode de GAUSS et de JACOBI. Pour chacune de ces methodes , la comparaison des courbes des erreurs avec celles trouvees avant la minimisation montre clairement que l'amelioration est nette.



SOLUTION PAR METHODE DE GAUSS

1

	0,42836	0,19537	-0,05009	
0,53931		0,31427	0,07341	-0,17556
	0,53836	0,31688	0,07610	
0,64067		0,43090	0,19801	-0,05077
	0,63984	0,43347	0,20094	
0,73191		0,54075	0,31963	0,07481
	0,72939	0,54084	0,31963	
0,81002		0,63996	0,43353	0,19655
	0,80799	0,64013	0,43353	
0,87595		0,72959	0,54095	0,31582
	0,87550	0,73129	0,54291	
0,92954		0,81952	0,64198	0,43179
	0,92932	0,81109	0,64388	
0,93853		0,87684	0,73305	0,54144



ERREUR PAR METHODE DE GAUSS (en  $10^{-3}$ )

1

	0,266	0,162	-0,679	
0,156		0,243	-0,039	-0,451
	0,356	0,264	-0,074	
0,201		0,339	0,123	0,062
	0,422	0,454	0,379	
0,223		0,451	0,430	0,336
	0,500	0,543	0,432	
0,264		0,536	0,513	0,317
	0,641	0,708	0,508	
0,340		0,701	0,650	0,337
	0,548	0,534	0,323	
0,287		0,560	0,461	0,216
	0,573	0,531	0,271	
0,300		0,575	0,433	0,179

SOLUTION PAR METHODE DE JACOBI

1

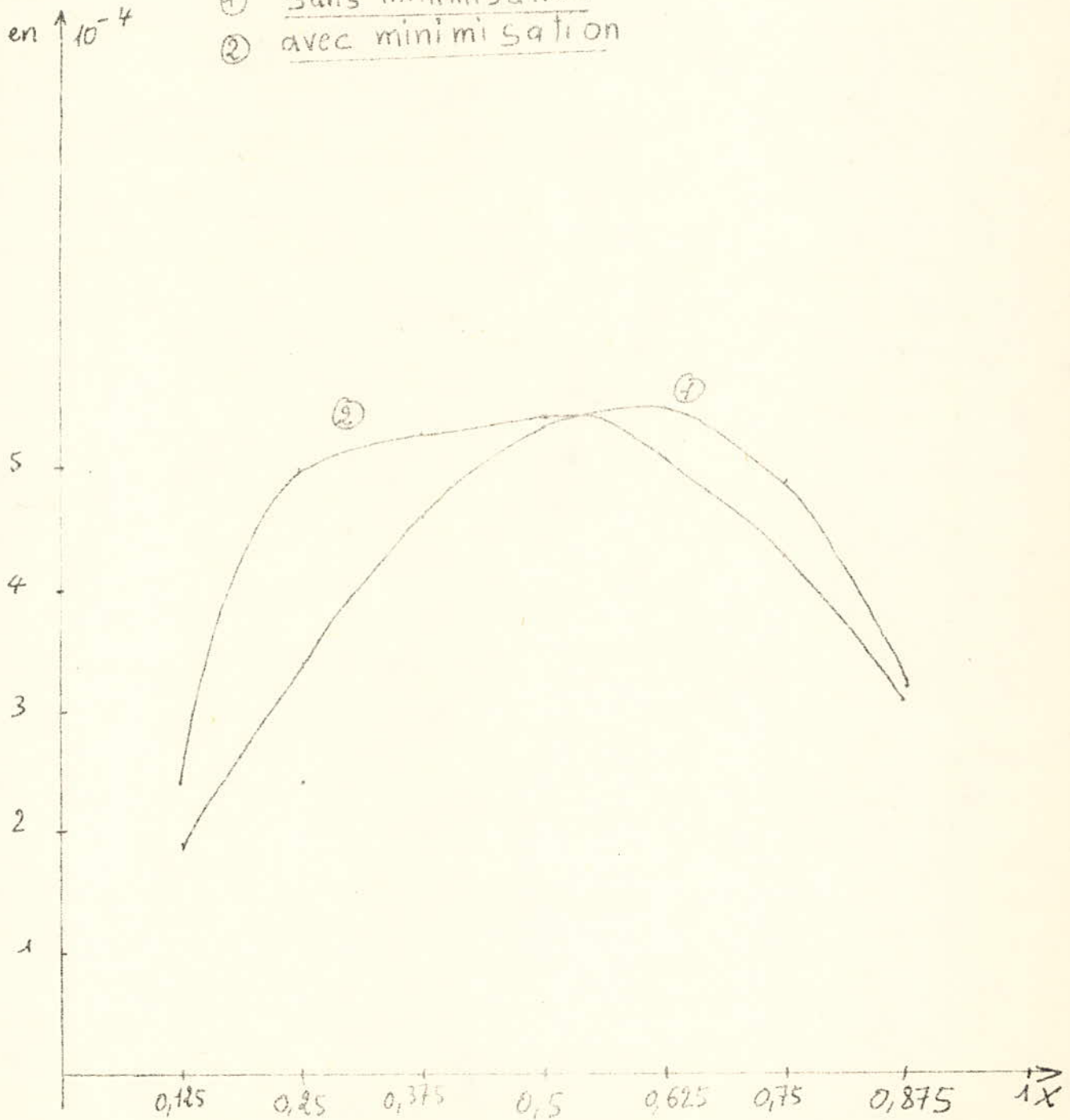
	0,42836	0,19537	-0,05000	
0,53931	0,31427	0,07341	-0,17601	
	0,53236	0,31688	0,07610	
0,64067	0,43090	0,19801	-0,05073	
	0,63984	0,43347	0,20094	
0,73191	0,54075	0,31963	0,07515	
	0,72939	0,54084	0,31963	
0,81003	0,63996	0,43353	0,19687	
	0,80800	0,64013	0,43353	
0,87595	0,72959	0,54095	0,31566	
	0,87550	0,73129	0,54291	
0,92954	0,80952	0,64198	0,43200	
	0,92932	0,81109	0,64388	
0,96853	0,87684	0,73305	0,54163	

ERREUR PAR JACOBI ( en  $10^{-3}$  )

	0,265	0,161	-0,079	
0,156		0,243	-0,039	-0,451
	0,335	0,264	-0,074	
0,201		0,338	0,123	0,042
	0,423	0,454	0,378	
0,223		0,451	0,430	0,336
	0,502	0,542	0,431	
0,265		0,536	0,512	0,317
	0,642	0,706	0,507	
0,341		0,700	0,650	0,337
	0,548	0,534	0,322	
0,287		0,560	0,460	0,216
	0,574	0,532	0,270	
0,301		0,575	0,483	0,179

ERREURS OBTENUES PAR METHODE DE GAUSS  
Pour  $\gamma = 0.5$

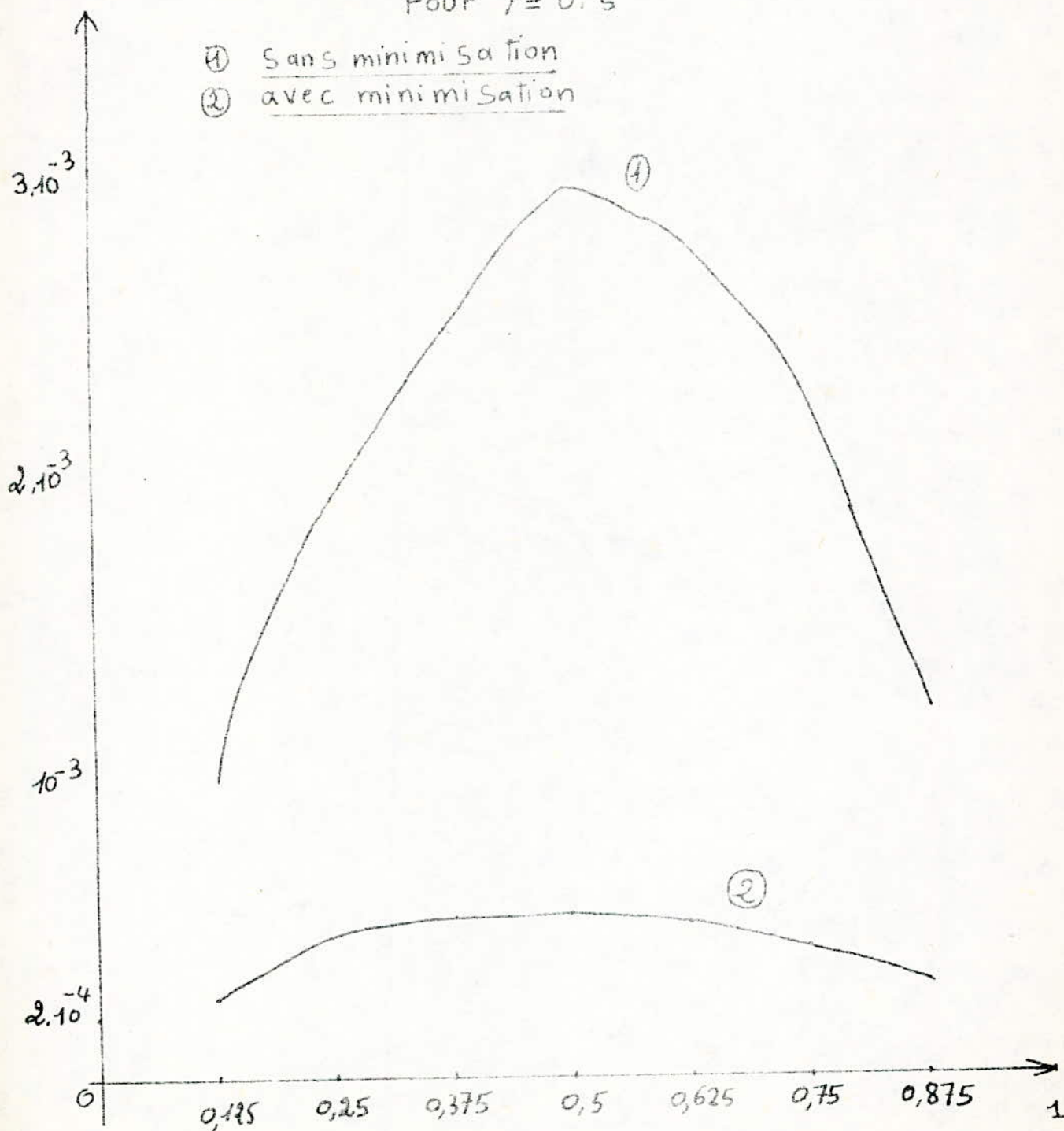
- ① Sans minimisation
- ② avec minimisation



# ERREURS OBTENUES PAR METHODE DE JACOBI

Pour  $Y = 0,5$

- ① Sans minimisation
- ② avec minimisation



E - CONCLUSION

Notre travail a porte essentiellement sur l'elaboration d'une methode generale de l'uniformisation de l'erreur due a la resolution des equations aux derivees partielles par des methodes iteratives. L'illustration de cette methode par une application simple (cas de deux maillages avec comme parametre  $R=0$  ) donne de bons resultas. La courbe de l'erreur par JACOBI le montre bien.

Le temps qui nous a ete imparti au passage des programmes sur ordinateur ne suffisait pas pour pouvoir faire davantage d'applications. Notamment dans le cas ou le parametre  $R$  serait different de zero. Ceci nous aurait permis verifier notre methode dans un cas tres general (pour le cas de quatre blocs) et d'evaluer le cout de l'operation. Cette derniere partie est tres importante pour la rentabilite de notre methode.

Nous esperons, par notre travail, avoir fourni les elements necessaires a la mise au point d'une methode generale de l'uniformisation de l'erreur due a la resolution des equations aux derivees partielle.

\*\*\*\*\*  
\* BIBLIOGRAPHIE \*  
\*\*\*\*\*  
.....

- THEORIE DES MATRICES

F.R. GANTMACHER ( tome 1 )

-MATRICES NONNEGATIVES ET NORMES VECTORIELLES

F.ROBERT ( INP GRENOBLE )

- MATRIX ITERATIVE ANALYSIS PRENTICE HALL

R.S. VARGA

- SOLUTIONS NUMERIQUES DES EQUATIONS ALGEBRIQUES

E. DURAND (tome 2 )