

N° d'ordre : 869

50376  
1980  
206

50376  
1980  
206

UNIVERSITÉ DES SCIENCES ET TECHNIQUES DE LILLE

# THÈSE

présentée à

L'UNIVERSITÉ DES SCIENCES ET TECHNIQUES DE LILLE

pour obtenir le titre de

**DOCTEUR DE 3<sup>ème</sup> CYCLE**

(Mathématiques Appliquées)

par

Yvon HAUW



**SUR DES APPLICATIONS  
DE LA METHODE DES PARAMETRES**



Thèse soutenue le Vendredi 12 décembre 1980 devant la Commission d'Examen

Membres du Jury	MM. P. POUZET	Président
	P. HUARD	Rapporteur
	C. BREZINSKI	Examineur
	J. DENEL	Examineur

P R O F E S S E U R S I E R E C L A S S E

M. BACCHUS Pierre	Mathématiques
M. BEAUFILS Jean Pierre (dét.)	Chimie
M. BIAYS Pierre	G.A.S.
M. BILLARD Jean	Physique
M. BONNOT Ernest	Biologie
M. BOUGHON Pierre	Mathématiques
M. BOURIQUET Robert	Biologie
M. CELET Paul	Sciences de la Terre
M. COEURE Gérard	Mathématiques
M. CONSTANT Eugène	I.E.E.A.
M. CORDONNIER Vincent	I.E.E.A.
M. DEBOURSE Jean Pierre	S.E.S.
M. DELATTRE Charles	Sciences de la Terre
M. DURCHON Maurice	Biologie
M. ESCAÏS Bertrand	Physique
M. FAURE Robert	Mathématiques
M. FOURET René	Physique
M. GABILLARD Robert	I.E.E.A.
M. GRANELLE Jean Jacques	S.E.S.
M. GRUSON Laurent	Mathématiques
M. GUILLAUME Jean	Biologie
M. HECTOR Joseph	Mathématiques
M. HEUBEL Joseph	Chimie
M. LABLACHE COMBIER Alain	Chimie
M. LACOSTE Louis	Biologie
M. LANSRAUX Guy	Physique
M. LAVEINE Jean Pierre	Sciences de la Terre
M. LEBRUN André	C.U.E.E.P.
M. LEHMANN Daniel	Mathématiques
Mme LENOBLE Jacqueline	Physique
M. LHOMME Jean	Chimie
M. LOMBARD Jacques	S.E.S.
M. LOUCHEUX Claude	Chimie
M. LUCQUIN Michel	Chimie
M. MAILLET Pierre	S.E.S.
M. MONTREUIL Jean	Biologie
M. PAQUET Jacques	Sciences de la Terre
M. PARREAU Michel	Mathématiques
M. PROUVOST Jean	Sciences de la Terre

Professeurs 1ère classe (suite)

M. SALMER Georges	I.E.E.A.
Mme SCHWARTZ Marie Hélène	Mathématiques
M. SEGUIER Guy	I.E.E.A.
M. STANKIEWICZ François	Sciences Economiques
M TILLIEU Jacques	Physique
M. TRIDOT Gabriel	Chimie
M. VIDAL Pierre	I.E.E.A.
M. VIVIER Emile	Biologie
M. WERTHEIMER Raymond	Physique
M. ZEYTOUNIAN Radyadour	Mathématiques

P R O F E S S E U R S 2ème C L A S S E

M. AL FAKIR Sabah	Mathématiques
M. ANTOINE Philippe	Mathématiques
M. BART André	Biologie
Mme BATTIAU Yvonne	Géographie
M. BEGUIN Paul	Mathématiques
M. BELLET Jean	Physique
M. BKOUCHE Rudolphe	Mathématiques
M. BOBE Bernard	S.E.S.
M. BODART Marcel	Biologie
M. BOILLY Bénoni	Biologie
M. BONNELLE Jean Pierre	Chimie
M. BOSQ Denis	Mathématiques
M. BREZINSKI Claude	I.E.E.A.
M. BRUYELLE Pierre (Chargé d'enseignement)	Géographie
M. CAPURON Alfred	Biologie
M. CARREZ Christian	I.E.E.A.
M. CHAMLEY Hervé	E.U.D.I.L.
M. CHAPOTON Alain	C.U.E.E.P.
M. COQUERY Jean Marie	Biologie
Mme CORSIN Paule	Sciences de la Terre
M. CORTOIS Jean	Physique
M. COUTURIER Daniel	Chimie
Mle DACHARRY Monique	Géographie
M. DEBRABANT Pierre	E.U.D.I.L.
M. DEGAUQUE Pierre	I.E.E.A.
M. DELORME Pierre	Biologie
M. DEMUNTER Paul	C.U.E.E.P.
M. DE PARIS Jean-Claude	Mathématiques

M. DEVRAINNE Pierre	Chimie
M. DHAINAUT André	Biologie
M. DORMARD Serge	S.E.S.
M. DOUKHAN Jean Claude	E.U.D.I.L.
M. DUBOIS Henri	Physique
M. DUBRULLE Alain	Physique
M. DUEE Gérard	Sciences de la Terre
M. DYMENT Arthur	Mathématiques
M. FLAMME Jean Marie	E.U.D.I.L.
M. FONTAINE Hubert	Physique
M. GERVAIS Michel	S.E.S.
M. GOBLOT Rémi	Mathématiques
M. GOSSELIN Gavriel	S.E.S.
M. GOUDMAND Pierre	Chimie
M. GREVET Patrice	S.E.S.
M. GUILBAULT Pierre	Biologie
M. HANGAN Théodor	Mathématiques
M. HERMAN Maurice	Physique
M. JACOB Gérard	I.E.E.A.
M. JACOB Pierre	Mathématiques
M. JOURNEL Gérard	E.U.D.I.L.
M. KREMBEL Jean	Biologie
M. LAURENT François	I.E.E.A.
Mlle LEGRAND Denise	Mathématiques
Mlle LEGRAND Solange	Mathématiques (Calais)
Mme LEHMANN Josiane	Mathématiques
M. LEMAIRE Jean	Physique
M. LENTACKER Firmin	G.A.S.
M. LEVASSEUR Michel	I.P.A.
M. LHENAFF René	G.A.S.
M. LOCQUENEUX Robert	Physique
M. LOSFELD Joseph	I.E.E.A.
M. LOUAGE Francis	E.U.D.I.L.
M. MACKE Bruno	Physique
M. MAIZIERES Christian	I.E.E.A.
Mlle MARQUET Simone	Mathématiques
M. MESSELYN Jean	Physique
M. MIGEON Michel	E.U.D.I.L.
M. MIGNOT Fulbert	Mathématiques
M. MONTEL Marc	Physique
Mme NGUYEN VAN CHI Régine	G.A.S.
M. PARSY Fernand	Mathématiques
Mlle PAUPARDIN Colette	Biologie

Professeurs 2ème classe (suite)

M. PERROT Pierre	Chimie
M. PERTUZON Emile	Biologie
M. PONSOLLE Louis	Chimie
M. PORCHET Maurice	Biologie
M. POVY Lucien	E.U.D.I.L.
M. RACZY Ladislas	I.E.E.A.
M. RICHARD Alain	Biologie
M. RIETSCH François	E.U.D.I.L.
M. ROGALSKI Marc	M.P.A.
M. ROUSSEAU Jean-Paul	Biologie
M. ROY Jean-Claude	Biologie
M. SALAMA Pierre	S.E.S.
Mme SCHWARZBACH Yvette (CCP)	M.P.A.
M. SCHAMPS Joël	Physique
M. SIMON Michel	S.E.S.
M. SLIWA Henri	Chimie
M. SOMME Jean	G.A.S.
Mle SPIK Geneviève	Biologie
M. STERBOUL François	E.U.D.I.L.
M. TAILLIEZ Roger	Institut Agricole
M. TOULOTTE Jean-Marc	I.E.E.A.
M. VANDORPE Bernard	E.U.D.I.L.
M. WALLART Francis	Chimie
M. WATERLOT Michel	Sciences de la Terre
Mme ZINN JUSTIN Nicole	M.P.A.

CHARGES DE COURS

M. TOP Géard	S.E.S.
M. ADAM Michel	S.E.S.

CHARGES DE CONFERENCES

M. DUVEAU Jacques	S.E.S.
M. HOFACK Jacques	I.P.A.
M. LATOUCHE Serge	S.E.S.
M. MALAUSSENA DE PERNO Jean-Louis	S.E.S.
M. OPIGEZ Philipe	S.E.S.

Monsieur Le Professeur POUZET a bien voulu me faire l'honneur de présider le Jury ; je le prie de croire à ma profonde gratitude.

Je sais gré à Monsieur Le Professeur HUARD de m'avoir donné l'idée de ce travail et d'en avoir suivi l'évolution. Je suis heureux de pouvoir lui exprimer ici toute ma reconnaissance pour les nombreux conseils qu'il n'a cessé de me prodiguer.

Que Monsieur Jacques DENEL, qui m'a toujours encouragé dans ce travail, veuille trouver ici l'expression de ma reconnaissance.

Je remercie Monsieur Le Professeur Claude BREZINSKI qui a bien voulu s'intéresser à ce travail et le juger.

Je remercie également les membres de l'Equipe d'Optimisation du Laboratoire de Mathématiques Appliquées et tout particulièrement Gérard PLATEAU, pour l'aide qu'ils m'ont apportée tout au long de ce travail.

J'adresse enfin mes remerciements pour la qualité de la frappe et la réalisation matérielle à Mesdames NIREL et TAILLY et à Monsieur et Madame DEBOCK dont j'ai pu apprécier la compétence et la gentillesse.

A la mémoire de mon Père.

# PLAN

## INTRODUCTION

### CHAPITRE I : RAPPELS

	Page
INTRODUCTION	1
A/ Rappel des principales méthodes directes de résolution de systèmes d'équations linéaires	
A-1 : La méthode de Gauss	2
A-2 : La méthode de Jordan	4
A-3 : Autres méthodes	4
B/ Application au cas des matrices creuses	
B-1 : Définition et situation du problème	6
B-2 : Différentes solutions	7

### CHAPITRE II : LA MÉTHODE DES PARAMÈTRES

INTRODUCTION	1
A/ Présentation de la méthode	
A-1 : Origine	2
A-2 : Le principe de résolution	2
A-3 : Le nombre d'opérations	3
A-4 : Comparaison avec d'autres méthodes	6
A-5 : Existence et choix du pivot	6
A-6 : Résultats numériques	8
A-7 : La RCL Factorisation	16
B/ Le cas des matrices creuses	
B-1 : L'idée de base	18
B-2 : Le but et les moyens de la méthode	21
B-3 : Phase 1 : Rangement	22
B-4 : Phase 2 : Résolution	49
B-5 : La RCL factorisation	44
C/ Adaptation à quelques problèmes particuliers	48



## CHAPITRE III : APPLICATIONS À LA MÉTHODE SIMPLICIALE

INTRODUCTION	1
A/ La méthode simpliciale et les différentes résolutions	
A-1 : Rappel de la méthode simpliciale	2
A-2 : Les différentes résolutions	3
B/ Cas de la méthode des paramètres	
B-1 : Résolution directe des systèmes linéaires	6
B-2 : Remarques concernant l'adaptation	7
C/ Expériences et résultats numériques	
C-1 : Les conditions d'expériences	10
C-2 : Les résultats numériques	12
C-3 : Remarques et commentaires	25
 Annexe 1 : Matrices test	
 Annexe 2 : Code de programmation	
 Bibliographie	

## ERRATA

### Chapitre II

- Page 2 : ligne 14 en un système  
Page 6 : ligne 5 ou  
Page 7 : ligne 10  $(A \ Q) \neq 0$   
Page 19 : ligne 10 relation  
Page 21 : ligne 21 en 2 phases  
Page 22 : ligne 4 ne comportent pas  
Page 27 : ligne 8 expériences  
Page 37 : ligne 2 sur lesquelles  
Page 42 : ligne 14 projectifs  
Page 47 : ligne 11 car seules

### Chapitre III

- Page 2 : ligne 19 définit  
          ligne 21  $I' = I - r + s$   
Page 7 : ligne 10 n'importe quelle  
Page 11 : ligne 33 langage

## INTRODUCTION

Les principes de résolution des systèmes d'équations linéaires sont connus depuis longtemps, ces derniers tenant dans les mathématiques une place fort importante.

La transcription de ces principes à l'outil informatique, qui a donnée naissance aux algorithmes de résolution (chapitre I-A), a introduit, en plus de la contrainte "temps de résolution", une nouvelle contrainte : "la place mémoire", principalement dans le cas de systèmes de grande taille.

C'est à partir de ces deux contraintes qu'ont été développées les applications au cas des matrices creuses (chapitre I-B).

La Méthode des paramètres, conçue par P. HUARD pour répondre aux mêmes considérations, et l'algorithme correspondant viennent ainsi enrichir l'ensemble des méthodes existantes tant dans le principe (chapitre II-A) que dans l'application (chapitre II-B).

C'est bien sûr dans cette dernière que la méthode peut prouver sa valeur.

Aussi l'avons nous utilisée dans le cadre de la méthode simpliciale (chapitre III-A), son adaptation ne présentant aucune difficulté (chapitre III-B) et les résultats (chapitre III-C) pouvant justifier l'intérêt de la méthode des paramètres.

# CHAPITRE I

## RAPPELS .

## INTRODUCTION

Les méthodes de résolution des systèmes d'équations linéaires sont nombreuses, aussi ne présentons-nous dans ce chapitre que certaines des méthodes de résolution directes. Deux raisons expliquent ce choix :

- Elles mettent en évidence les différents principes de base (section A).
- Ce sont celles employées dans le cas des matrices creuses (section B).



existence : basée sur le théorème suivant

(théorème III - [1p75])

Etant donnée une matrice  $A \in M(n,n)$ , il existe toujours une matrice de permutation  $P$  telle que  $A_1 = P.A = TA'$  avec  $T$  triangulaire inférieure unitaire et  $A'$  triangulaire supérieure.

Les nombres  $a_{kk}^{(k)}$  sont appelés les pivots. On doit les supposer tous différents de zéro. De plus, leur choix est essentiel en ce qui concerne la stabilité de l'algorithme. Aussi plusieurs variantes ont-elles été développées.

- pivotage partiel (en ligne ou en colonne) : option L ou C  
on recherche le "meilleur" pivot à l'étape  $(k+1)$  sur la ligne ou la colonne  $k$ , puis on permute les lignes ou les colonnes correspondantes.

- pivotage total : option L C  
on recherche le "meilleur" pivot dans la sous-matrice  $B = \{A_{ij} / i, j \geq k\}$  et on permute les lignes et les colonnes correspondantes.

Remarques :

Comme il n'existe pas de définition du "meilleur" pivot, plusieurs critères ont été étudiés et utilisés.

On peut prendre pour pivot :

- l'élément de valeur absolue maximum
- l'élément dont la valeur se rapproche le plus de la moyenne des valeurs (Durand)
- celui qui se rapproche le plus de  $|\det A|^{\frac{1}{n}}$  [2p217].

Il apparait en tout cas nécessaire de prévoir une recherche du pivot si l'on veut que la méthode soit stable numériquement.

On pourra consulter à ce sujet les références [1] à [4] .

## A-2- La méthode de Jordan

Principe : transformer le système  $Ax = b$  en un système équivalent  $\overset{(n)}{A} x = \overset{(n)}{b}$  où  $\overset{(n)}{A}$  est une matrice diagonale.

$$\overset{(k)}{A} = \begin{array}{ccc} \begin{array}{c} (1) \\ a_{11} \end{array} & \begin{array}{c} 0 \\ \vdots \\ a_{22} \\ \vdots \\ 0 \end{array} & \begin{array}{c} (k) \\ a_{1k} \\ \vdots \\ a_{kk} \\ \vdots \\ a_{nk} \end{array} \\ & & \begin{array}{c} (k) \\ \vdots \\ a_{kn} \\ \vdots \\ a_{nn} \end{array} \end{array} \quad \begin{array}{c} (k) \\ b_1 \\ \vdots \\ b_k \\ \vdots \\ b_n \end{array} = \begin{array}{c} (k) \\ b \\ \vdots \\ b \end{array}$$

formules de transformation :

$$\overset{(k+1)}{a}_{ij} = \overset{(k)}{a}_{ij} - \frac{\overset{(k)}{a}_{ik} \overset{(k)}{a}_{kj}}{\overset{(k)}{a}_{kk}} \quad \begin{array}{l} i = 1, n \quad i \neq k \\ j = k+1, n \end{array}$$

$$\overset{(k+1)}{b}_i = \overset{(k)}{b}_i - \frac{\overset{(k)}{a}_{ik} \overset{(k)}{b}_k}{\overset{(k)}{a}_{kk}} \quad i = 1, n \quad i \neq k$$

existence : basée sur le théorème suivant  
(théorème I [1P90])

Etant donnée  $A$  non singulière, il existe une matrice  $S$  telle que  $\det S = +1$  et  $S.A. = D$  avec  $D$  matrice diagonale.

De même que dans la méthode de Gauss, le problème de la recherche du pivot se pose et les mêmes conclusions sont valables.

variante : la méthode de Jordan "exacte" [3P25]

## A-3- Autres méthodes

Nous ne ferons ici que citer les autres méthodes. En effet, leur exposé ne mettrait en évidence aucun nouveau principe de base, celles-ci ayant pour but de triangulariser la matrice de départ à l'aide de transformations particulières.



- Méthode de Housolder : les matrices de passage de  $A^{(k)}$  à  $A^{(k+1)}$  sont des matrices orthogonales.

- Méthode d'orthogonalisation de Schmidt : la matrice triangulaire supérieure obtenue est "orthogonale en ligne"

- Méthode de rotations de Givens : on transforme le système initial à l'aide de  $n(n-1)/2$  matrices de rotations, voir [1].

Ces méthodes, si elles ont l'avantage d'être stables numériquement, sont cependant très onéreuses en temps de calcul.

Ainsi :

- Méthode de Housolder : [4p21]

$$\frac{4n^3 + 9n^2 - n - 12}{6} \quad \text{additions}$$

$$\frac{2n^3 + 6n^2 + 4n - 12}{3} \quad \text{multiplications}$$

$$\frac{n^2 + 3n - 2}{2} \quad \text{divisions}$$

(n-1) racines carrées.

- Méthode de Schmidt [1p100]

$$n(n+2)(n-1) \quad \text{additions}$$

$$\frac{n(2n^2 - 3n + 1)}{2} \quad \text{multiplications}$$

$$n^2 \quad \text{divisions}$$

- Méthode de Givens [2p245]

$$\frac{4}{3} n^3 \quad \text{multiplications}$$

$$\frac{1}{2} n^2 \quad \text{racines carrées}$$

Ces volumes de calculs, beaucoup plus importants que ceux des méthodes de Gauss et Jordan, font que ces méthodes sont inemployées dans le cas des systèmes à matrices creuses.

## B/ APPLICATION AU CAS DES MATRICES CREUSES

### B-1- Définition et situation du problème

Dans de nombreux domaines, -électricité, théorie des graphes, équations différentielles-, les matrices des systèmes d'équations linéaires à résoudre comportent une grande partie d'éléments nuls.

Aussi le cas des systèmes à matrice creuse a-t-il été particulièrement étudié afin de tenir compte, lors de la résolution, des gains de temps et de place mémoire rendus possibles par le creux des matrices (opérations et mémorisation ne portant que sur des éléments non nuls).

#### Définitions :

soit  $n$  la dimension du système  $Ax = b$

soit  $N$  le nombre d'éléments non nuls de la matrice  $A$

\*  $\rho = \frac{N}{n^2}$  est appelée la densité de la matrice  $A$

\*  $(1-\rho)$  est appelé le creux de la matrice  $A$

Le tableau suivant donne les valeurs de  $\rho$  généralement admises comme caractérisant des matrices creuses en fonction de la dimension du système [5].

n	50	100	500	1000	5000
$\rho$	25%	10%	2%	0,5%	0,1%

#### Remarque :

On ne peut prévoir qu'un élément sera nul que s'il est obtenu :

- soit par addition ou soustraction de deux zéros
- soit par produit d'un zéro par un autre élément
- soit par division d'un zéro par un élément non nul.

Situation du problème :

exemple [6p30]

soit la matrice A d'un système  $Ax = b$

$$A = \begin{array}{c} \begin{array}{cccc} & 1 & 2 & 3 & 4 \\ 1 & X & X & X & X \\ 2 & X & X & & \\ 3 & X & & X & \\ 4 & X & & & X \end{array} \end{array}$$

Si on résoud le système par la méthode de Gauss en prenant successivement pour pivot  $a_{11}, a_{22}, a_{33}$ , on obtient :

$$\begin{array}{c} \begin{array}{c} (2) \\ A = \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \\ X & X & X & X \\ & X & X & X \\ & X & X & X \\ & X & X & X \end{array} \end{array} \quad \begin{array}{c} \begin{array}{c} (3) \\ A = \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \\ X & X & X & X \\ & X & X & X \\ & & X & X \\ & & X & X \end{array} \end{array} \quad \begin{array}{c} \begin{array}{c} (4) \\ A = \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \\ X & X & X & X \\ & X & X & X \\ & & X & X \\ & & & X \end{array} \end{array} \end{array}$$

La matrice triangulaire résultante est entièrement remplie de termes non nuls.

Si on choisit successivement pour pivot  $a_{44}, a_{33}, a_{22}$  (dans les équations 4, 3, 2,), on obtient

$$\begin{array}{c} \begin{array}{c} (2) \\ A = \end{array} \begin{array}{c} \begin{array}{cccc} 4 & 3 & 2 & 1 \\ X & & & X \\ & X & & X \\ & & X & X \\ & & X & X & X \end{array} \end{array} \quad \begin{array}{c} \begin{array}{c} (3) \\ A = \end{array} \begin{array}{c} \begin{array}{cccc} 4 & 3 & 2 & 1 \\ X & & & X \\ & X & & X \\ & & X & X \\ & & X & X \end{array} \end{array} \quad \begin{array}{c} \begin{array}{c} (4) \\ A = \end{array} \begin{array}{c} \begin{array}{cccc} 4 & 3 & 2 & 1 \\ X & & & X \\ & X & & X \\ & & X & X \\ & & & X \end{array} \end{array} \end{array}$$

(4)

La matrice A ne compte alors que 7 éléments non nuls.

Il est donc important de bien "choisir" l'ordre d'utilisation des équations et dans celles-ci les pivots ; c'est le point de départ de toutes les adaptations des méthodes de résolution au cas des systèmes à matrices creuses.

## B-2- Différentes solutions

La liste des différentes solutions présentée ci-dessous n'est pas exhaustive. Comme on l'a vu précédemment, elles

ont pour but de répondre à un ou plusieurs des trois critères :

- gain de temps (opérations non effectuées sur des éléments nuls),
- gain de place (mémorisation d'éléments non nuls)
- précision des résultats (choix du pivot).

Ce dernier critère étant pris souvent en compte implicitement.

Dans la littérature très abondante, relative au sujet qui nous intéresse, il nous a paru intéressant de classer les méthodes en deux groupes et ce par rapport aux buts essentiels de la méthode :

groupe 1 : celles dont le but est le gain de temps, objectif réalisé en évitant que la matrice ne se remplisse de termes non nuls au cours des différentes itérations de la résolution.

groupe 2 : celles dont les buts sont des gains de temps et de place mémoire, objectifs réalisés en tenant compte d'une structure particulière de la matrice.

### B-2-1 Méthodes du groupe 1 [6] à [13]

Toutes ces méthodes diffèrent par le critère déterminant l'ordre d'utilisation des équations

- règle 1 :

Les lignes de A sont préordonnées avant l'élimination, classées en fonction du nombre de termes non nuls qu'elles contiennent. La première ligne est la plus creuse.

- règle 2 :

A chaque étape de l'élimination, on choisit comme ligne du pivot la ligne la plus creuse de la matrice.

- règle 3 :

A chaque étape de l'élimination, on choisit la ligne du pivot de façon que la transformation effectuée crée le plus petit nombre d'éléments non nuls  $P_k$

3 variantes possibles

- dans la sous-matrice  $i > k, j > k$
- dans la sous-matrice  $i \geq k, j > k$
- dans la sous-matrice  $i \geq k, j \geq k$

Les pivots sont les éléments de la diagonale principale.

- règle 4 :

on choisit comme pivot l'élément, qui minimise  $(r_i - 1)(C_j - 1)$   
où  $r_i$  et  $c_j$  sont les degrés des ligne i et colonne j.

- règle 5 :

on choisit comme pivot l'élément appartenant à la colonne de degré minimum.

- règle 6 :

on choisit comme pivot l'élément appartenant à la colonne qui minimise  $S_j$  (nombre total d'éléments appartenant aux lignes i telles que  $A_i^j \neq 0$ ).

- règle 7 :

on choisit comme pivot l'élément appartenant à la colonne qui minimise

$$\gamma_j = (S_j - C_j) (C_j - 1) / C_j$$

Les méthodes 4 à 7 sont en fait des méthodes portant sur le choix des pivots, en accord avec un critère numérique sur la valeur des éléments. Il va de soi qu'elles peuvent être combinées avec une méthode portant sur le choix des lignes. On peut aussi citer dans ce groupe 1 les méthodes des articles [12] et [13], celles de ce dernier s'appliquent à des matrices symétriques en structure.

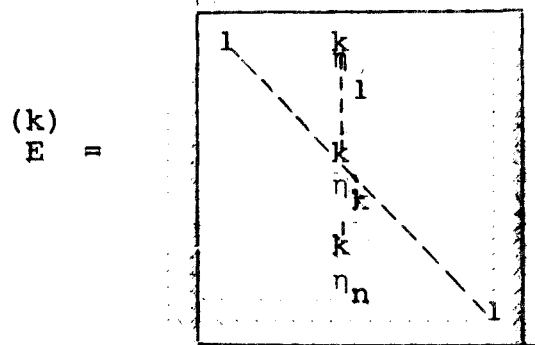
#### B-2-2- Méthode du groupe 2

Elles consistent à mettre A sous une forme particulière, en structure ou en mémorisation, ce qui permet une localisation facile des éléments non nuls, dont on tiendra compte lors de la résolution.

#### forme particulière en structure :

Quelques formes les plus couramment rencontrées sont représentées ci-après :





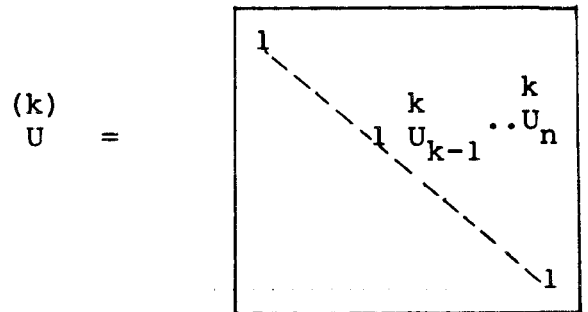
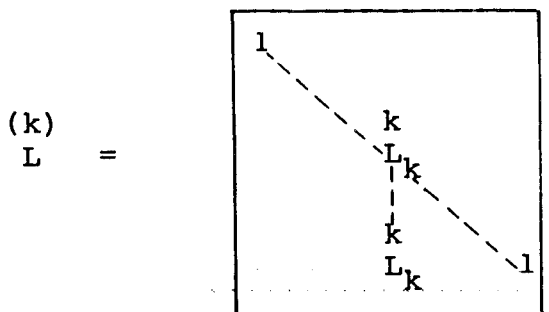
et  $\eta^k = E^{(k-1)-1} \dots E^{(2)-1} E^{(1)-1} A^k$

La factorisation EFI.

La matrice A est factorisée sous la forme  $A = LU$ , avec L triangulaire inférieure et U triangulaire supérieure ; les Matrices L et U sont elles-mêmes factorisées sous la forme d'un produit de matrices élémentaires unicolonnes pour L et unilignes pour U.

$A \neq LU$  avec

$L = L^{(1)} L^{(2)} \dots L^{(n)}$  et  $U = U^{(1)} U^{(2)} \dots U^{(n)}$  où



La Bi-factorisation [22].

Basée sur l'équation

$$\begin{matrix} (n) & & (2) & (1) & & (1) & (2) & & (n) \\ L & \dots & L & L & A & R & R & \dots & R = I \end{matrix}$$

Les matrices  $L^{(k)}$  et  $R^{(k)}$  sont de la même forme que les matrices  $L^{(k)}$  et  $U^{(k)}$  précédentes.



## CHAPITRE II

### LA METHODE DES PARAMETRES

## INTRODUCTION

La méthode des paramètres est une méthode de résolution directe des systèmes d'équations linéaires par diagonalisation, qui entre dans le même cadre que les méthodes présentées au chapitre I (section A).

Sa caractéristique essentielle est d'être une méthode qui s'adapte "facilement" au cas des systèmes à matrice creuse, en tenant compte du creux de la matrice et des obligations de mémorisation des résultats intermédiaires de la résolution (section B).

Enfin, aucune hypothèse préalable n'étant nécessaire sur la structure de la matrice de départ, son adaptation à certains cas particuliers n'en est que plus intéressante (section C).

## A/ PRÉSENTATION DE LA MÉTHODE

### A-1- Origine

La méthode, due à P. HUARD [23], est en fait la généralisation à tous les systèmes à matrice creuse d'un principe utilisé jusqu'alors pour certains types très particuliers de matrices, comme dans les problèmes de flot avec gains par J.F. MAURRAS [24].

Cependant, avant de la présenter dans son cadre d'origine - les matrices creuses - il convient d'en donner les principes de base afin de pouvoir la comparer aux méthodes exposées précédemment.

### A-2- Le principe de résolution.

Soit un système  $Ax = b$ .

Principe : transformer le système  $Ax = b$  est un système équivalent

$$\begin{matrix} (2n) & (2n) & (2n) \\ A & x & = & b \end{matrix} \text{ où } A \text{ est la matrice identité.}$$

Pour cela, on calcule une suite de matrices

$$A = \begin{matrix} (1) & (2) & (3) & \dots & (2n) \\ A & , & A & , & A \end{matrix} = I \text{ et}$$

$$b = \begin{matrix} (1) & (2) & (3) & \dots & (2n) \\ b & , & b & , & b \end{matrix} = x \text{ telles que}$$

$(2k-1)$   
 $A$   
 $1 \leq k \leq n$

$1$ $1$	$(2k-1)$ $a_{1,k}$	$(2k-1)$ $a_{1,k+1}$
$0$ ----- $0$	$(2k-1)$ $a_{k,k}$	$(2k-1)$ $a_{k,k+1}$
$(2k-1)$ $a_{k+1,k}$	$(2k-1)$ $a_{n,k}$	

$(2k+1)$ $b_1$
$(2k+1)$ $b_k$
$(2k-1)$ $b_n$

$=$   $(2k-1)$   
 $b$

$(2k)$   
 $A$   
 $1 \leq k \leq n$

$1$ $1$	$0$ $0$	$(2k)$ $a_{1,k+1}$
$0$ ----- $0$	$1$	$(2k)$ $a_{k,k+1}$
$(2k)$ $a_{k+1,k}$	$(2k)$ $a_{n,k}$	

$(2k)$ $b_1$
$(2k)$ $b_k$
$(2k)$ $b_{k+1}$
$(2k)$ $b_n$

$=$   $(2k)$   
 $b$



Formules de transformation.

$$\begin{matrix} (2k-1) \\ a_{ij} \end{matrix} = \begin{matrix} (2k-2) \\ a_{ij} \end{matrix} \quad \text{pour } j=1, n \text{ et } i=1, (k-1)$$

$$\begin{matrix} (2k-1) \\ a_{ij} \end{matrix} = \begin{matrix} (2k-2) \\ a_{ij} \end{matrix} = \begin{matrix} (1) \\ a_{ij} \end{matrix} \quad \text{pour } j=1, n \text{ et } i=(k+1), n$$

$$\begin{matrix} (2k-1) \\ a_{kj} \end{matrix} = 0 \quad \text{pour } j=1, (k-1)$$

$$\begin{matrix} (2k-1) \\ a_{kj} \end{matrix} = \begin{matrix} (2k-2) \\ a_{kj} \end{matrix} - \sum_{l=1}^{k-1} \begin{matrix} (2k-2) \\ a_{kl} \end{matrix} \cdot \begin{matrix} (2k-2) \\ a_{lj} \end{matrix} \quad \text{pour } j=k, n$$

$$\begin{matrix} (2k-1) \\ b_i \end{matrix} = \begin{matrix} (2k-2) \\ b_i \end{matrix} \quad \text{pour } i=1, n \quad i \neq k$$

$$\begin{matrix} (2k-1) \\ b_k \end{matrix} = \begin{matrix} (2k-2) \\ b_k \end{matrix} - \sum_{l=1}^{k-1} \begin{matrix} (2k-2) \\ a_{kl} \end{matrix} \cdot \begin{matrix} (2k-2) \\ b_l \end{matrix}$$

$$\begin{matrix} (2k) \\ a_{ij} \end{matrix} = \begin{matrix} (2k-1) \\ a_{ij} \end{matrix} \quad \text{pour } j=1, (k-1) \text{ et } i=1, k$$

$$\begin{matrix} (2k) \\ a_{ij} \end{matrix} = \begin{matrix} (2k-1) \\ a_{ij} \end{matrix} = \begin{matrix} (1) \\ a_{ij} \end{matrix} \quad \text{pour } j=1, n \text{ et } i=(k+1), n$$

$$\begin{matrix} (2k) \\ a_{ik} \end{matrix} = 0 \quad \text{pour } i=1, (k-1)$$

$$\begin{matrix} (2k) \\ a_{ij} \end{matrix} = \begin{matrix} (2k-1) \\ a_{ij} \end{matrix} - \frac{\begin{matrix} (2k-1) \\ a_{ik} \end{matrix} \begin{matrix} (2k-1) \\ a_{kj} \end{matrix}}{\begin{matrix} (2k-1) \\ a_{kk} \end{matrix}} \quad \text{pour } j=(k+1), n \text{ et } i=1, (k-1)$$

$$\begin{matrix} (2k) \\ a_{kj} \end{matrix} = \frac{\begin{matrix} (2k-1) \\ a_{kj} \end{matrix}}{\begin{matrix} (2k-1) \\ a_{kk} \end{matrix}} \quad \text{pour } j=k, n$$

$$\begin{matrix} (2k) \\ b_i \end{matrix} = \begin{matrix} (2k-1) \\ b_i \end{matrix} \quad \text{pour } i=(k+1), n \quad \begin{matrix} (2k) \\ b_k \end{matrix} = \frac{\begin{matrix} (2k-1) \\ b_k \end{matrix}}{\begin{matrix} (2k-1) \\ a_{kk} \end{matrix}}$$

$$\begin{matrix} (2k) \\ b_i \end{matrix} = \begin{matrix} (2k-1) \\ b_i \end{matrix} - \frac{\begin{matrix} (2k-1) \\ a_{ik} \end{matrix} \begin{matrix} (2k-1) \\ b_i \end{matrix}}{\begin{matrix} (2k-1) \\ a_{kk} \end{matrix}} \quad \text{pour } i=1, (k-1)$$

A-3 - Le nombre d'opérations.

On voit, d'après les formules de transformation, qu'il faut effectuer :

- pour passer de  $A^{(2k-2)}$  à  $A^{(2k+1)}$  :  
 $(k-1)(n-k+2)$  multiplications et additions

- pour passer de  $A^{(2k-1)}$  à  $A^{(2k)}$  :  
 $(n-k+1)$  divisions  
 $(k-1)(n-k+1)$  multiplications et additions

soit, pour la résolution du système :

:	:	$\frac{n(n-1)(2n+5)}{6}$	:
:	: multiplications ou additions :		:
:	:	6	:
:	:		:
:	:	$\frac{n(n+1)}{2}$	:
:	: divisions :		:
:	:	2	:

Démonstration :

- le nombre de divisions est égal à :

$$\sum_{k=1}^n (n-k+1) = n(n+1) - \sum_{k=1}^n k = n(n+1) - \frac{n(n+1)}{2} = \frac{n(n+1)}{2}$$

- le nombre de multiplications et d'additions est égal à :

$$S = \sum_{k=1}^n [(k-1)(n-k+2) + (k-1)(n-k+1)] = \sum_{k=1}^n (k-1)(2n-2k+3) =$$

$$\sum_{k=1}^n (2nk-2n-2k^2+5k-3) = (2n+5) \sum_{k=1}^n k - 2 \sum_{k=1}^n k^2 - 2n^2 - 3n$$

$$\text{or } \sum_{k=1}^n k^2 = \frac{n(n+1)(2n+1)}{6} \quad \text{d'où}$$

$$S = (2n+5) \left( \frac{n(n+1)}{2} \right) - \frac{2n(n+1)(2n+1)}{6} - 2n^2 - 3n$$

$$\text{d'où } 6S = 2n^3 + 3n^2 - 5n = n(n-1)(2n+5)$$

$$\text{soit } S = \frac{n(n-1)(2n+5)}{6}$$

#### A-4- Comparaison avec d'autres méthodes

- on obtient, pour la méthode de Gauss [1 p82]

	triangularisation	Résol. syst.	Total
multiplications et additions	$n(n-1)/3$	$n(n-1)/2$	$n(n-1)(2n+5)/6$
divisions	$n(n-1)/2$	$n$	$n(n+1)/2$

- de même, pour la méthode de Jordan [1p92]

multiplications ou additions	$(n-1) \left[ \frac{n(n+1)}{2} - 1 \right]$
divisions	$n^2$

Ainsi, la méthode des paramètres, qui est une méthode de diagonalisation comme la méthode de Jordan, nécessite le même nombre d'opérations que la méthode de triangularisation de Gauss.

#### A-5 - Existence et choix du pivot.

Existence : les résultats donnés jusqu'à présent nécessitent l'hypothèse que tous les pivots rencontrés  $a_{kk}^{(2k-1)}$  ( $k=1, n$ ) sont non nuls. On va montrer, à l'aide des théorèmes qui suivent, qu'ils restent valables dans le cas contraire car on peut alors trouver une permutation des colonnes de  $A^{(2k-1)}$  telle que  $a_{kk}^{(2k-1)}$  devienne non nul.

Théorème :  $A$  inversible  $\iff \forall k, \exists Q$  telle que  $(A \ Q)_{kk} \neq 0$   
 $1 \leq k \leq (n-1)$ ,  $Q$  étant une matrice de permutation.

Démonstration : condition nécessaire

La négation :  $(\exists k$  tel que  $\forall Q, (A \ Q)_{kk} = 0)$  signifie que  
 $a_{kj} = 0$  pour  $j = k, n$ . Comme  $a_{kj} = 0$  pour  $j=1, (k-1)$ , on a  
 $a_{kj} = 0 \ \forall j$ .

La ligne  $k$  est donc une combinaison linéaire des lignes  
 $1$  à  $(k-1)$ ,  $A$  n'est pas inversible.

condition suffisante

$\forall k, \exists Q$  telle que  $(A \ Q)_{kk} \neq 0$ . On a alors  $A^{-1} = I$ , donc le produit  
des matrices de transformation (voir la factorisation RCL) est égal  
à  $A^{-1}$ .

Théorème :  $A$  inversible  $\implies (a_{nn} \neq 0)$

Démonstration : même démonstration que pour l'implication ( $\implies$ ) du  
théorème précédent avec  $Q = I$  et  $k=n$ .

Choix du pivot :

On vient de voir que si  $A$  est inversible, il existe  
toujours une matrice de permutation (notée  $Q$ ) telle que le pivot  
soit non nul. Cependant, même si  $a_{kk}$  est différent de zéro  $\forall k$ ,  
on peut choisir les pivots, à l'aide de matrices de permutation,  
qui conduisent à de "bons" résultats du point de vue numérique.  
On retrouve là les options (L), (C) et (LC) de la méthode de Gauss  
(chapitre I).

Pour la méthode des paramètres, seule une recherche du  
pivot sur les colonnes est possible à l'étape  $k$ , car les lignes  
 $(k+1)$  à  $n$  n'ont pas encore été modifiées pour tenir compte de  
l'expression des  $(k-1)$  premières variables déjà exprimées. Pré-  
cisons que dans les expériences numériques décrites plus loin,



- I le critère retenu de sélection du pivot est toujours le maximum des valeurs absolues.

#### A-6- Résultats numériques

Les expériences décrites ci-dessous ont pour but de vérifier que la méthode des paramètres ne donne pas des résultats systématiquement meilleurs ou moins bons que la méthode de Gauss avec recherche du pivot dans la sous matrice "restante" - option "pivotage total" -

Deux raisons justifient ces expériences :

- la méthode des paramètres n'autorise que la recherche du pivot sur les colonnes
- Au cours de la résolution d'un système, on ne commet aucune erreur sur les éléments de la matrice situés sous la diagonale car ils sont inchangés (au signe près) par la méthode des paramètres.

#### Les conditions d'expériences

- 3 types de matrices ont été choisis :
  - . matrice de Rutherford : très bon conditionnement
  - . matrice de Milnes paramétrée :
    - C = 1 très bon conditionnement
    - C = 0,001 conditionnement moyen
  - . matrice de Durand n°2 : mauvais conditionnement
- pour chaque type, N (ordre du système) varie :
  - . 10,20,30,40,50 (Rutherford et Milnes)
  - . 6,8,10,12 (Durand)
- Deux systèmes de même ordre (matrices types différentes) ont même solution.
- Deux systèmes d'ordres différents (même matrice type ou pas) ont même solution pour les (Min {ordres}) premières composantes.
- Les solutions ont été générées par une procédure au hasard avec même initialisation, ce qui a permis de calculer les seconds membres.

## Présentation des résultats

. Pour chacun des tableaux ci-dessous :

- N indique l'ordre du système qui a été résolu et dont les résultats apparaissent dans la colonne correspondante.

-  $\| \cdot \|_1$  signifie :  $\|x\|_1 = \sum_{i=1}^N |x_i|$

-  $\| \cdot \|_\infty$  signifie :  $\|x\|_\infty = \sup_{1 \leq i \leq N} |x_i|$

- EA signifie erreur absolue : solution "exacte" - solution calculée

- ER signifie erreur relative

- G signifie : résultat par la méthode de Gauss - "pivotage total"

- P signifie : résultat par la méthode des paramètres -  
"pivotage sur les colonnes"

- P<sub>1</sub> (matrice de Durand) signifie : résultat par la méthode des paramètres sans pivotage

EA		ER	
$\  \cdot \ _1$ G	$\sum_{i=1}^N  (X_{\text{exact}} - X_{\text{Gauss}})_i $	$\  \cdot \ _1$ G	$\sum_{i=1}^N \left  \frac{(X_{\text{exact}} - X_{\text{Gauss}})_i}{(X_{\text{exact}})_i} \right $
$\  \cdot \ _1$ P	$\sum_{i=1}^N  (X_{\text{exact}} - X_{\text{para}})_i $	$\  \cdot \ _1$ P	$\sum_{i=1}^N \left  \frac{(X_{\text{exact}} - X_{\text{para}})_i}{(X_{\text{exact}})_i} \right $
$\  \cdot \ _1$ P <sub>1</sub>	$\sum_{i=1}^N  (X_{\text{exact}} - X_{\text{para SP}})_i $	$\  \cdot \ _1$ P <sub>1</sub>	$\sum_{i=1}^N \left  \frac{(X_{\text{exact}} - X_{\text{para SP}})_i}{(X_{\text{exact}})_i} \right $
$\  \cdot \ _\infty$ G	$\sup_{1 \leq i \leq N}  (X_{\text{exact}} - X_{\text{Gauss}})_i $	$\  \cdot \ _\infty$ G	$\sup \left  \frac{(X_{\text{exact}} - X_{\text{Gauss}})_i}{(X_{\text{exact}})_i} \right $
$\  \cdot \ _\infty$ P	$\sup_{1 \leq i \leq N}  (X_{\text{exact}} - X_{\text{para}})_i $	$\  \cdot \ _\infty$ P	$\sup \left  \frac{(X_{\text{exact}} - X_{\text{para}})_i}{(X_{\text{exact}})_i} \right $
$\  \cdot \ _\infty$ P <sub>1</sub>	$\sup_{1 \leq i \leq N}  (X_{\text{exact}} - X_{\text{para SP}})_i $	$\  \cdot \ _\infty$ P <sub>1</sub>	$\sup \left  \frac{(X_{\text{exact}} - X_{\text{para SP}})_i}{(X_{\text{exact}})_i} \right $

Matrice de Rutherford - Très bon conditionnement -

Matrice du type :

$$A = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 3 & 3 & \dots & 3 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & 3 & 5 & \dots & 5 \end{bmatrix} \quad a_{i,j} = 2 \cdot \min(i,j) - 1$$

Résultats en  $10^{-14}$ 

		N	20	30	40	50
EA	$\  \cdot \ _1 G$		1	52	127	219
	$\  \cdot \ _1 P$		2	47	114	236
	$\  \cdot \ _\infty G$		0,2	9	13,8	13,5
	$\  \cdot \ _\infty P$		0,1	8,8	11,7	11,7
ER	$\  \cdot \ _1 G$		42	219	604	896
	$\  \cdot \ _1 P$		6,7	153	571	963
	$\  \cdot \ _\infty G$		38	77	155	160
	$\  \cdot \ _\infty P$		1,2	38	155	160



Matrice de Milnes paramétrée · c=1 - très bon conditionnement.

Matrice du type :

$$A = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1+c & 1 & 1 & \dots & 1 \\ \vdots & 3 & 1 & \dots & \vdots \\ \vdots & \vdots & 4 & \dots & \vdots \\ 1+c & 3 & 4 & \dots & n & 1 \end{bmatrix}$$

lorsque C diminue, le conditionnement devient moins bon.

Résultats en  $10^{-14}$

		N	10	20	30	40	50
EA	$\  \cdot \ _1 G$		0,08	0,2	0,5	1,8	2,8
	$\  \cdot \ _1 P$		0,3	1,5	2	4,8	9,7
	$\  \cdot \ _\infty G$		0,03	0,07	0,1	0,5	0,8
	$\  \cdot \ _\infty P$		0,1	0,4	0,5	2,3	3,2
ER	$\  \cdot \ _1 G$		0,2	0,5	2	6	7
	$\  \cdot \ _1 P$		4	32	60	73	80
	$\  \cdot \ _\infty G$		0,1	0,2	0,9	2,3	2,5
	$\  \cdot \ _\infty P$		3	28	48	57	51

Matrice de Milnes paramétrée :  $c=0,001$  - conditionnement moyen.

Résultats en  $10^{-14}$

		N	30	40	50
EA	$     _1 G$		460	266	900
	$     _1 P$		37	231	300
	$     _\infty G$		230	132	450
	$     _\infty P$		18	114	150
ER	$     _1 G$		4740	1420	4250
	$     _1 P$		440	1300	1550
	$     _\infty G$		3250	850	2800
	$     _\infty P$		260	730	920



Matrice de Durand n° 2 - mauvais conditionnement -

Matrice du type :

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 3 & 4 & \dots & \\ 1 & 3 & 6 & 10 & \dots & \\ 1 & 4 & 10 & 20 & \dots & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & & & & & \end{bmatrix}$$

$$a(1,j) = 1 \quad 1 \leq j \leq n$$

$$a(i,1) = 1 \quad 1 \leq i \leq n$$

$$a(i,j) = a(i,j-1) + a(i-1,j)$$

Résultats en  $10^{-12}$ ,  $10^{-10}$ ,  $10^{-8}$ ,  $10^{-6}$

		N	$6_{(10^{-12})}$	$8_{(10^{-10})}$	$10_{(10^{-8})}$	$12_{(10^{-6})}$
EA	}	$\  \  \  \ _1 G$	1,5	2	10	6
		$\  \  \  \ _1 P$	2	6	4,5	5
		$\  \  \  \ _1 P1$	1,8	7	27	190
		$\  \  \  \ _\infty G$	0,5	0,6	2,7	1,4
		$\  \  \  \ _\infty P$	0,6	1,7	1,1	1,1
		$\  \  \  \ _\infty P1$	0,6	2	6,8	42
ER	}	$\  \  \  \ _1 G$	95	135	430	145
		$\  \  \  \ _1 P$	140	400	180	105
		$\  \  \  \ _1 P1$	110	470	1080	3900
		$\  \  \  \ _\infty G$	90	130	410	130
		$\  \  \  \ _\infty P$	130	390	175	100
		$\  \  \  \ _\infty P1$	105	450	1030	3500

Remarque :

Lors de la résolution par la méthode des paramètres, les options P (recherche du pivot sur les colonnes) et P1 (pas de recherche sur le pivot) n'ont pu être comparées que dans le cas de la matrice de Durand n°2. En effet, les matrices de Rutherford et de Milnes étant telles que  $A_{ij} = k_i$  pour  $j \geq i$ , une recherche du pivot sur les colonnes aurait été inutile.

Conclusion.

- Matrice de Rutherford : les résultats sont équivalents tant en erreur absolue qu'en erreur relative.

- Matrice de Milnes  $c=1$  : la méthode de Gauss est toujours meilleure avec un rapport égal à 4 pour l'erreur absolue et 20 pour l'erreur relative

- Matrice de Milnes  $c=0,0001$  : la méthode des paramètres est toujours meilleure, avec un rapport variant de 10 à 3 (en passant par 1,2 pour  $N=40$ ) quelque soit l'erreur.

- Matrice de Durand n° 2 : la méthode des paramètres, option P1, est toujours la plus mauvaise et peut conduire à des résultats désastreux ( $N = 10,12$ ).

La méthode de Gauss est meilleure pour  $N = 6$  et  $8$  et moins bonne pour  $N = 10$  et  $12$  que la méthode des paramètres, option P.

Il ressort de ces expériences numériques que :

1°) si on devait classer les matrices par rapport à leur conditionnement (du meilleur au moins bon), on obtiendrait

(pour un même ordre des matrices)

Matrice de Milnes paramétrée  $c=1$

Matrice de Rutherford

Matrice de Milnes paramétrée  $c=0,001$

Matrice de Durand n° 2.

2°) La méthode des paramètres, option P, semble moins sensible au conditionnement de la matrice du système que la méthode de Gauss

Le tableau suivant donne, pour chacune des méthodes, normes et types d'erreur, la valeur du rapport de la borne supérieure et de la borne inférieure de la plage de variation des erreurs à N constant sur les trois premières matrices.

		EA			ER		
		30	40	50	30	40	50
<sub>1</sub>	Milnes						
	Gauss	920	150	320	2370	236	607
	Para	24	48	31	7	18	19
<sub>∞</sub>	Gauss	2300	264	562	3611	350	1120
	Para	36	50	47	6	13	18



A-7- La RCL factorisation.

Basée sur l'équation matricielle suivante :

$$\begin{matrix} (n) & (n) & (n-1) & (n-1) & & (2) & (2) & (1) & (1) \\ C & L & C & L & \dots & C & L & C & L & A = I \end{matrix}$$

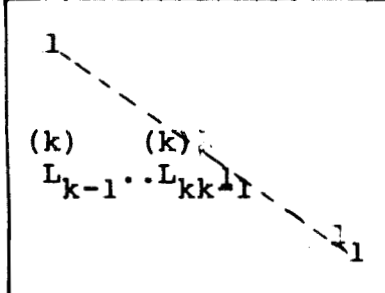
soit  $A^{-1} = \begin{matrix} (n) & (n) & (n-1) & (n-1) & & (1) & (1) \\ C & L & C & L & \dots & C & L \end{matrix}$

où  $\begin{matrix} (k) & (k-1) & & (1) & (1) & & (2k-1) \\ L & C & \dots & C & L & A = & A \end{matrix}$

et  $\begin{matrix} (k) & (k) & (k-1) & & (1) & (1) & & (k) & (2k-1) & (2k) \\ C & L & C & \dots & C & L & A = & C & A & = A \end{matrix}$

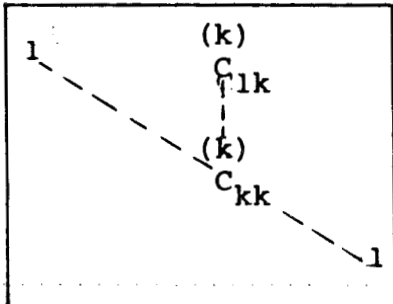
On obtient des matrices de la forme :

$\begin{matrix} (k) \\ L = \end{matrix}$



$\begin{matrix} (k) & (2k-2) & (1) \\ L_{kj} = -A_{kj} & = -A_{kj} \end{matrix}$   
 $1 \leq j \leq (k-1)$

$\begin{matrix} (k) \\ C = \end{matrix}$



$\begin{matrix} (k) & (2k-1) \\ C_{kk} = 1/A_{kk} \end{matrix}$   
 $\begin{matrix} (k) & (2k-1) & (2k-1) \\ C_{jk} = -A_{jk} / A_{kk} \end{matrix}$   
 $1 \leq j \leq (k-1)$

Démonstration.

$$\binom{(k)}{(L \quad A)}_{kj} = \sum_{i=1}^k \binom{(k)}{L_{ki}} \binom{(2k-2)}{A_{ij}} \text{ en particulier pour}$$

$j=1, (k-1)$  mais

$$\binom{(2k-2)}{A_{ij}} = 0 \quad \text{pour } j < k, i < k \text{ et } i \neq j$$

$$\binom{(2k-2)}{A_{jj}} = 1 \text{ pour } j < k$$

$$\binom{(k)}{L_{kk}} = 1$$

$$\text{donc } \binom{(k)}{(L \quad A)}_{kj} = \binom{(k)}{L_{kj}} + \binom{(2k-2)}{A_{kj}} = 0 \quad j=1, (k-1)$$

Comme  $\binom{(2k-2)}{A_{kj}} = \binom{(1)}{A_{kj}}$  car le produit de A par  $\binom{(k-1)}{C} \binom{(k-1)}{L} \dots \binom{(1)}{C}$  laisse inchangées les lignes k à n, on obtient

$$\binom{(k)}{L_{kj}} = - \binom{(1)}{A_{kj}} \text{ pour } k = 2 \text{ à } n, j = 1 \text{ à } (k-1).$$

Les formules donnant la valeur des éléments  $\binom{(k)}{C_{jk}}$  pour  $j \leq k$  sont évidentes par rapport aux formules de transformation de la pages 4. Chapitre II.

**Remarques :** - la matrice  $\binom{(1)}{L}$  est la matrice identité

- il n'y aura aucun calcul à effectuer pour trouver les matrices  $\binom{(k)}{L}$  ( $k=1, n$ ) ; seules les matrices  $\binom{(k)}{C}$  devront être calculées, d'où le nom de la méthode :

la rapide - colonne - ligne (RCL) factorisation

- le produit des pivots est égal au déterminant de A

$$|A^{-1}| = \binom{(n)}{C} \binom{(n)}{L} \dots \binom{(1)}{C} \binom{(1)}{L} = \binom{(n)}{C} \binom{(n)}{L} \dots \binom{(1)}{C} \binom{(1)}{L}$$

mais  $\binom{(k)}{L} = 1 \quad \forall k$  donc

$$|A^{-1}| = \binom{(n)}{C} \binom{(n-1)}{C} \dots \binom{(1)}{C} = \frac{1}{\binom{(2n)}{A_{nn}}} \times \frac{1}{\binom{(2n-2)}{A_{n-1, n-1}}} \times \dots \times \frac{1}{\binom{(2)}{A_{11}}} = \frac{1}{|A|}$$

$$\text{donc } |A| = \prod_{k=1}^n \binom{(2k)}{A_{kk}}$$

## B/ LE CAS DES MATRICES CREUSES

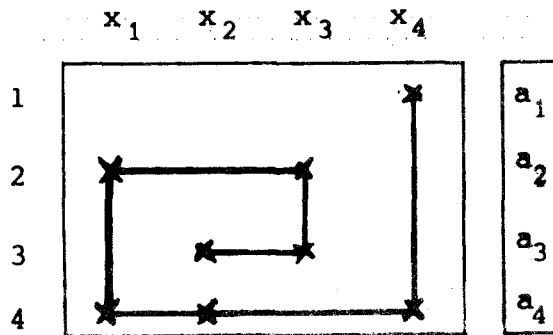
### B-1- L'idée de base

L'idée de P. HUARD[23] apparait comme la généralisation à toutes les matrices creuses de deux applications particulières relativement à la structure des matrices auxquelles elles s'appliquent.

#### 1 - Problèmes de flots avec gain - J.F. MAURRAS[24] -

Les matrices des systèmes à résoudre possèdent deux éléments non nuls par colonne

exemple :



On peut résoudre un tel système en calculant successivement

avec l'équation 1 :  $x_4 = a_1$

avec l'équation 4 :  $x_1 = f(x_2, x_4) = f(x_2)$   
 $x_4$  ayant été calculé à l'étape précédente

avec l'équation 2 :  $x_3 = f(x_1) = f(x_2)$   
 $x_1$  étant connu en fonction de  $x_2$

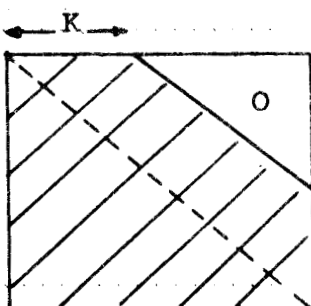
avec l'équation 3 :  $x_3 = f(x_2)$

Les deux dernières équations obtenues permettent de calculer  $x_3$  et  $x_2$  donc  $x_1$ .

On résout le système en introduisant une variable "paramétré", ici  $x_2$ .

## 2 - Les matrices "bandes"

Elles sont telles que



soit le système suivant

$$\begin{array}{|c|c|} \hline A_1^P & A_1^Q \quad O \\ \hline A_2^P & A_2^Q \\ \hline \end{array}
 \begin{array}{c} x_P \\ x_Q \end{array}
 =
 \begin{array}{c} a_1 \\ a_2 \end{array}$$

La résolution comporte trois phases :

- l'introduction de P paramètres

$$A_1^P x_P + A_1^Q x_Q = a_1 \quad \text{soit} \quad x_Q = (A_1^Q)^{-1} a_1 - (A_1^Q)^{-1} A_1^P x_P = b - B x_P$$

- le calcul de la valeur de ces P paramètres

$$A_2^P x_P + A_2^Q x_Q = a_2 \quad \text{soit} \quad [A_2^P - A_2^Q B] x_P = a_2 - A_2^Q b$$

- le calcul de  $x_Q$ , connaissant  $x_P$ , dans la première relation  $x_Q = b - B x_P$ .

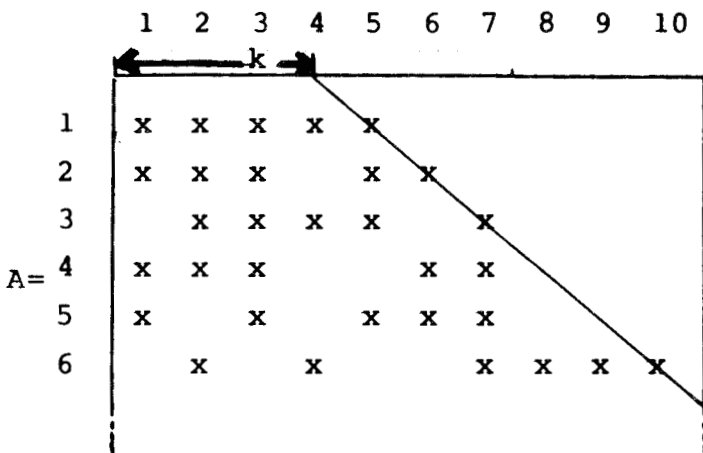
La Matrice  $A_1^Q$  doit être triangulaire inférieure inversible

Généralisation :

- les calculs s'effectuent équation par équation

- on introduit de nouveaux paramètres lorsque c'est nécessaire et on "élimine" des paramètres dès que possible.

Exemple 5 : soit un système  $Ax = b$



Nombre de paramètres :

- équation 1 : k-1
- équation 2 : k-1
- 3 : k-1
- 4 : k-2
- 5 : k-3
- 6 : k-1
- ⋮

maximum atteint :  $P = k-1$

le début de la résolution donnera :

équation 1 :

$$x_1 = f(x_2, x_3, x_4, x_5)$$

équation 2 :

$$x_6 = f(x_1, x_2, x_3, x_5) = f(x_2, x_3, x_4, x_5)$$

équation 3 :

$$x_7 = f(x_2, x_3, x_4, x_5)$$

équation 4 :

$$0 = f(x_1, x_2, x_3, x_6, x_7) = f(x_2, x_3, x_4, x_5) \rightarrow x_5 = f(x_2, x_3, x_4) \quad x_5 = f(x_2, x_3)$$

équation 5 :

$$0 = f(x_1, x_3, x_5, x_6, x_7) = f(x_2, x_3, x_4) \rightarrow x_4 = f(x_2, x_3)$$

équation 6 :

$$x_8 = f(x_2, x_4, x_7, x_8, x_9) = f(x_2, x_3, x_8, x_9)$$

$$x_1 = f(x_2, x_3, x_4) \quad x_4 = f(x_2, x_3)$$

$$\uparrow \quad \uparrow$$

$$x_6 = f(x_2, x_3, x_4) \quad x_6 = f(x_2, x_3)$$

$$\uparrow \quad \uparrow$$

$$x_7 = f(x_2, x_3, x_4) \quad x_7 = f(x_2, x_3)$$

$$\uparrow \quad \uparrow$$

$$x_5 = f(x_2, x_3, x_4) \quad x_5 = f(x_2, x_3)$$

$$\uparrow$$

$$x_4 = f(x_2, x_3)$$

⋮

De cette manière, on utilise les équations 1, une après l'autre pour :

- soit exprimer une variable en fonction des variables considérées comme paramètres à cette étape (équations 1,2,3,6)

- soit exprimer un paramètre en fonction des autres

variables paramètres (équations 4 et 5). Dans ce cas, on recal-  
cule l'expression des variables déjà exprimées par rapport à ce  
paramètre en fonction des paramètres restants. C'est la phase dite  
"d'élimination d'un paramètre".

#### B-2- Le but et les moyens de la méthode

Le but de la méthode est de permettre la résolution

de systèmes d'équations linéaires à matrice creuse et de

grande taille en mémoire centrale d'ordinateur. Cela implique :

- de ne mettre en mémoire centrale d'ordinateur que les éléments  
non nuls de la matrice initiale,

- d'avoir, en mémoire centrale, suffisamment de place pour

ranger les résultats des calculs intermédiaires de la résolution.

Il faut donc connaître, avant de résoudre le système,

le volume de place nécessaire, et pouvoir accéder facilement

aux différents éléments apparaissant en cours de résolution.

Pour toutes ces raisons, la méthode se décompose sur 2 phases :

phase 1 : "rangement de la matrice ou détermination de l'ordre

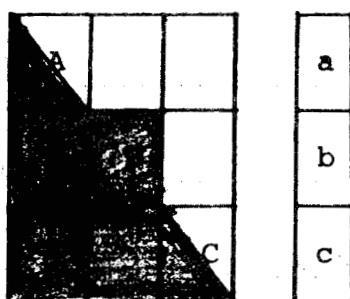
d'utilisation des équations

phase 2 : résolution du système

B-3- Phase 1 : "rangement"B.3.1. : Hypothèses préalables

- On supposera, dans ce paragraphe, que les matrices des systèmes à résoudre ne comporte pas de lignes ou de colonnes de degré 1 car, alors, le problème se pose sur un sous-système qu'on peut extraire.

exemple :



Il suffira de résoudre le problème pour la matrice B car la résolution de  $Ax_A = a$  est immédiate ainsi que celle de  $Cx_C = c$  lorsque  $x_A$  et  $x_B$  sont déterminés.

- On ne détruit pas la matrice du système à résoudre.

B-3-2- Position du problème. Le nombre maximum de paramètres

- comme on l'a vu précédemment, il s'agit de déterminer et, si possible, de minimiser par la première phase de la méthode, le volume de place mémoire nécessaire pour la résolution du système, celui-ci étant directement lié à l'ordre d'utilisation des équations.

Mathématiquement, on recherche une permutation des lignes et des colonnes de la matrice A et le nombre  $M_0$  de mémoires correspondant nécessaire pour la résolution.

Il faut cependant remarquer que pour une permutation donnée, plusieurs valeurs de  $M_0$  peuvent permettre de résoudre le système, ceci sans rien changer aux principes de calcul de la méthode des paramètres.

Exemple : soit la matrice d'un système obtenue après permutation des lignes et des colonnes.

	1	2	3	4	5	6	7
1	X	X	X				
2	X	X	X				
3	X	X	X	X	X	X	
4	X	X	X	X	X	X	
5	X	X	X	X	X	X	X
6	X	X	X	X	X	X	X
7	X	X	X	X	X	X	X

Les différentes étapes de la résolution donneront :

$$\begin{array}{l}
 \text{étape 1 : } x_1 = f(x_2, x_3) \quad x_1 = f(x_2) \quad x_1 \\
 \text{étape 2 : } 0 = f(x_2, x_3) \rightarrow x_3 = f(x_2) \quad \uparrow \quad x_3 \\
 \text{étape 3 : } \quad \quad \quad x_4 = f(x_2, x_5, x_6) \quad x_4 = f(x_2, x_5) \quad x_4 = f(x_2) \quad x_4 \\
 \text{étape 4 : } \quad \quad \quad 0 = f(x_2, x_5, x_6) \rightarrow x_6 = f(x_2, x_5) \quad x_6 = f(x_2) \quad x_6 \\
 \text{étape 5 : } \quad \quad \quad \quad \quad \quad x_7 = f(x_2, x_5) \quad x_7 = f(x_2) \quad x_7 \\
 \text{étape 6 : } \quad \quad \quad \quad \quad \quad \quad \quad \quad 0 = f(x_2, x_5) \rightarrow x_5 = f(x_2) \quad x_5 \\
 \text{étape 7 : } \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad 0 = f(x_2) \rightarrow x_2
 \end{array}$$

Le nombre maximum de paramètres étant égal à 3, un tableau auxiliaire de dimensions  $(7 \times 3)$ , soit 21 mémoires, suffit à mémoriser les coefficients des variables paramètres dans les différentes fonctions calculées en cours de résolution.



Si on représente l'état de ce tableau auxiliaire aux différentes étapes de la résolution, on obtient :

étape 1

	2	3	
1	x	x	

étape 2

	2	3	
1	x	x	
?	x	x	

étape 3

	2	5	6
1	x		
3	x		
4	x	x	x

étape 4

	2	5	6
1	x		
3	x		
4	x	x	x
?	x	x	x

étape 5

	2	5	
1	x		
3	x		
4	x	x	
6	x	x	

étape 6

	2	5	
1	x		
3	x		
4	x	x	
6	x	x	
7	x	x	

	2	5	
1	x		
3	x		
4	x	x	
6	x	x	
7	x	x	
?	x	x	

	2		
1	x		
3	x		
4	x		
6	x		
7	x		
5	x		

étape 7

	2		
1	x		
3	x		
4	x		
6	x		
7	x		
5	x		
?	x		

1			
3			
4			
6			
7			
5			
2			

$b_1$
$b_3$
$b_4$
$b_6$
$b_7$
$b_5$
$b_2$

= b



Il apparait alors que le tableau auxiliaire n'est jamais rempli complètement ; aussi peut-on rechercher des valeurs de  $M_0$  plus petites.

1°) La valeur exacte de Mo.

C'est le nombre maximum de mémoires nécessaires simultanément. Pour l'exemple donné, il est égal à 10 et correspond à la partie hachurée du tableau.


Cependant, résoudre le système avec ce nombre Mo de mémoires nécessite une hypothèse :

on élimine toujours le paramètre qui a été introduit en dernier, ce qui suppose que les valeurs correspondantes des pivots sont toujours non nulles et que l'option "pivotage sur les colonnes" est inutilisée.

Sans cette hypothèse, on ne peut plus contrôler le remplissage du tableau.

Exemple : à l'étape 4, on "élimine" le paramètre  $x_2$  et non plus  $x_6$ .

	2	5	6		5	6
1	x			1	x	x
3	x			3	x	x
4	x	x	x	4	x	x
?	x	x	x	2	x	x

on a, à ce moment, 8 éléments dans la matrice et le nombre de mémoires nécessaires simultanément devient égal à 12.

De plus, cette solution réalisable théoriquement, présente l'inconvénient constitué par l'indiciage des éléments intermédiaires de calcul qui doivent pouvoir être retrouvés rapidement donc indicés en conséquence.

Pour pallier cet inconvénient, on est donc conduit à rechercher une borne supérieure de  $M_0$ .

### 2°) Deuxième solution.

On détermine la valeur

$$M_0 = p_1 + \sum_{j=2}^n \sup_{k=j-1}^j (p_k)$$

pour l'exemple donné :  $M_0 = 15$

On conserve l'hypothèse précédente mais les problèmes d'indiciage sont simplifiés, une table de correspondance "équations position de la 1ère variable paramètre" relative au tableau de résolution permettant un accès rapide aux différents éléments.


### 3°) Troisième solution.

On détermine la valeur

$$M_0 = \sup_{k=1, n} (p_k) + \sum_{j=2}^n \sup_{k \geq j-1} (p_k)$$

pour l'exemple :  $M_0 = 17$

Cette solution a l'avantage de ne plus nécessiter l'hypothèse préalable du -1-, donc de permettre une recherche sur la valeur du pivot. De plus, les problèmes d'indiciage sont résolus comme dans le -2-.


4°) Quatrième solution.

$$M_0 = n \times \sup_{1 \leq k \leq n} p_k = n \times p$$

Cette solution autorise une recherche sur le pivot et permet, en travaillant directement sur un tableau auxiliaire de dimensions  $(n \times p)$ , de simplifier les problèmes d'indiçage.

Remarque : ces quatre solutions étant présentées, il faut préciser que c'est la dernière qui sera toujours utilisée dans les expériences numériques décrites plus loin.

En effet, les deux premières s'excluent à cause des résultats obtenus dans les expériences numériques sur la matrice de Durand n° 2 -PAGE 13, chapitre II- et la troisième présente peu d'intérêt si on tient compte de l'hypothèse faite en B.3.1 -page 22, chapitre II.

B-3-3 Enoncé du problème et visualisation

- soient A la matrice d'un système linéaire  $n \times n$ ,
  - $N = 1, 2, \dots, n$  l'ensemble des indices de ligne,
  - f une fonction de permutation sur N et  $f^{-1}$  la fonction réciproque
  - $J_i = \{j/A_{ij} \neq 0\}$  l'ensemble des colonnes "agissantes" de la ligne i
- on définit

$$N_{f(i)} = \bigcup_{k=1}^{f(i)} J_{f^{-1}(k)}, \text{ soit } |N_{f(i)}| \text{ son cardinal}$$

problème : déterminer P tel que

$$P = \min_f \{ \max_{i \in N} (|N_{f(i)}| - f(i)) \}$$

Théorème : A inversible  $\implies$

$$\forall i, (|N_{f(i)}| - f(i)) \geq 0 \quad \forall i = 1, n$$

démonstration : supposons le contraire :

$$\exists f \text{ et } i \quad (|N_{f(i)}| - f(i)) < 0 \text{ d'où } |N_{f(i)}| < f(i)$$

on pourrait donc extraire un sous-système rectangulaire de dimension  $f(i) \times |N_{f(i)}|$  et la matrice A ne pouvant être de rang N est non inversible.

Pour visualiser le résultat du problème, il suffit d'en changer l'énoncé afin de se ramener au cas d'une matrice triangulaire bande,  $P$  étant égal à l'épaisseur de la bande.

$$P = \min_{f,g} \{ \max_{i,j} \{ (g(j) - f(i)) \alpha_{ij} \} / j \in N, i \in N \}$$

où  $g$  : fonction de permutation sur  $N$

$$\alpha_{ij} = \begin{cases} 0 & \text{si } A_{ij} = 0 \\ 1 & \text{sinon} \end{cases}$$

Exemple : soit

	1	2	3	4	5	6
1	x			x		
2		x	x		x	
3	x		x			
4	x	x	x	x		
5		x		x	x	x
6			x	x	x	x

on obtient  $P = 1$  pour

$$f : \{1, 2, 3, 4, 5, 6\} \Rightarrow \{1, 3, 4, 2, 5, 6\}$$

$$g : \{1, 2, 3, 4, 5, 6\} \Rightarrow \{1, 4, 3, 2, 5, 6\}$$

	1	4	3	2	5	6
1	x	x				
3	x		x			
4	x	x	x	x		
2			x	x	x	
5		x		x	x	x
6		x	x		x	x

B-3-4- Algorithme donnant la solution optimale

La solution optimale d'un tel problème existe évidemment, la matrice A ayant un nombre fini de lignes et de colonnes.

Il faut aussi remarquer qu'elle peut ne pas être unique.

exemple : la matrice précédente avec

$$f : \{1, 2, 3, 4, 5, 6\} \rightarrow \{3, 1, 4, 2, 5, 6\}$$

$$g : \{1, 2, 3, 4, 5, 6\} \rightarrow \{1, 3, 4, 2, 5, 6\}$$

Il n'existe pas, à notre connaissance, d'algorithme donnant une solution optimale du problème en un nombre prédéterminé d'étapes éventuellement après calcul de valeurs relatives à la structure de la matrice et qui ne soit pas basé sur la réduction du nombre de permutations possibles ( $n!$  au départ).

On peut, bien sûr, en imaginer un basé sur la remise en cause de la solution trouvée à l'étape précédente et prenant en compte la structure de chaque ligne.

Le nombre de permutations possibles diminue très rapidement mais reste quand même très élevé. Aussi, avons-nous imaginé l'algorithme suivant.

Algorithme des sous-suites permutable complètes (SSPC)

soit  $N = \{1, 2, \dots, n\}$  l'ensemble des indices lignes

définition :  $S_1$  est une sous-suite permutable complète de  $N$

si  $\forall j \in S_1$ , il existe une permutation  $\theta_j$  telle que :

$$\theta_j : \{1, 2, \dots, k\} \rightarrow \{\theta(1), \theta(2), \dots, \theta(k)\}$$

où  $-\theta(j)=1$

$$- |J_1 \cup N_{\theta(1)-1}| - |N_{\theta(1)-1}| \leq 1 \quad 1 \in S, 1 \neq j$$

et  $\forall t \notin S, t \in N \quad |J_t \cup J_j| - |J_t \cap J_j| \geq 2$

exemple de SSPC :

$$\begin{array}{cccc} & 1 & 2 & 3 & 4 \\ 1 & \left( \begin{array}{cccc} x & x & & \\ & x & x & \\ & & x & x \end{array} \right), & & \begin{array}{cccc} & 1 & 2 & 3 & 4 & 5 \\ 1 & \left( \begin{array}{cccc} x & x & x & \\ & x & x & x & \\ & & x & x & x \end{array} \right) \end{array} \end{array}$$

Contre-exemple.

$$\begin{array}{c}
 1 \quad 2 \quad 3 \quad 4 \quad 5 \\
 1 \left( \begin{array}{ccccc}
 x & x & & & \\
 & x & x & x & \\
 & & & x & x
 \end{array} \right) \\
 2 \\
 3
 \end{array}$$

pour  $j = 3$ , il n'existe pas de permutation répondant aux conditions.

Théorème : La décomposition de  $N$  en sous-suites permutable complètes est unique.

Démonstration :

soient  $S_1, S_2, \dots, S_{t_1}$  et  $S'_1, S'_2, \dots, S'_{t_2}$  deux décompositions différentes de  $N$

Il existe donc au moins un indice  $j_1$  n'appartenant pas aux mêmes SSPC

soit  $j_1 \in S_1$  et  $j_1 \in S'_2$

mais alors, la sous-suite  $S = (S_1 \cup S'_2)$  est une SSPC, donc  $S_1$  et  $S'_2$  ne sont pas complètes d'où contradiction avec l'hypothèse.

Algorithme de recherche d'une solution optimale.

Les SSPC étant déterminées, on recherche une solution optimale de la manière suivante :

A l'étape  $k$ . 1°) existe-t-il des SSPC donnant un nombre de paramètres égal à  $k$  ?

non : on va en 2

oui : on les prend comme nouvelle suite de SSPC

2°) On complète les suites de SSPC existantes de telle manière que le nombre de paramètres ne dépasse jamais  $k$

Théorème : si à l'étape  $k$ , une suite de SSPC les utilise toutes, c'est une solution optimale.

La démonstration est évidente par la construction même des suites de SSPC.





solution optimale : P = 2

		1	2	3	4	5	6	7	10	8	9
1	9	x	x								
	5		x	x							
	1		x	x	x						
2	2	x				x	x				
	8					x	x	x			
4	4		x	x	x		x		x		
	6			x	x		x		x	x	
3	10				x					x	x
	3		x		x					x	x
	7			x						x	x

Ainsi, le nombre de permutations possibles est-il considérablement réduit au départ, ici de 10! à 4!

Il reste cependant que la recherche des SSPC d'une matrice demande beaucoup de temps, ce qui est en contradiction avec le but de la méthode. Aussi a-t-on été amené à tester, pour cette première phase, des heuristiques.

B-3-5- Les Heuristiques

Ne pouvant obtenir facilement la solution optimale du problème, on se fixe un critère de choix de l'ordre des lignes, ce qui permettra de déterminer la permutation f. On sélectionne les lignes une à une.

Notation : à l'étape (k), (k) lignes et (k+p<sub>k</sub>) colonnes auront été sélectionnées déterminant les sous-ensembles L<sub>k</sub> et C<sub>k</sub> tels que

$$A = \begin{array}{c} L_k \\ C \\ L_k \end{array} \begin{array}{cc} C_k & C_k^c \\ \hline & S_k \end{array}$$

On notera à l'étape k :

1-  $L_k$  l'ensemble des indices de ligne sélectionnés aux étapes 1 à (k-1)

$L_k^C$  son complémentaire sur N

$$C_k = \{j / A_{ij} \neq 0, i \in L_k\}$$

$C_k^C$  son complémentaire sur N

2-  $S_k = \{A_{ij} / i \in L_k^C, j \in C_k\}$  la sous matrice "restante"

3-  $d_i^{(k)}$  le degré de la ligne  $i \in L_k^C$  dans  $S_k$  et  $d_j^{(k)}$  le degré de la colonne  $j \in C_k^C$

4-  $J_i^{(k)} = \{j \in C_k^C / A_{ij} \neq 0\}$  soit  $|J_i^{(k)}| = d_i^{(k)}$

5-  $d_{i,t}^{(k+1)}$  le degré de la ligne  $i \in L_{k+1}^C$  après avoir sélectionné la ligne t à l'étape k.

### Les différents critères.

Pour les critères décrits ci-dessous, le choix d'une ligne à l'étape (k) est une ligne de degré minimum dans la matrice  $S_k$ . Ils diffèrent dans le cas où plusieurs lignes sont candidates.

Ainsi, quel que soit le critère, on sélectionne une ligne i appartenant à :

$$I_k = \{i \in L_k^C / d_i^{(k)} = \min(d_t^{(k)} / t \in L_k^C)\}.$$

et au cas où cet ensemble possède deux éléments ou plus :

Critère 1 : la dernière ligne candidate rencontrée

$$i = \max \{l / l \in I_k\}$$

Critère 2 : la première ligne candidate rencontrée

$$i = \min \{ l / l \in I_k \}$$

Critère 3 : celle possédant la colonne de degré maximum

$i$  telle que  $A_{ij} \neq 0$  pour  $j$  vérifiant

$$\bar{d}^j = \max \{ d^{(k)}_t / t \in \cup_{s \in J_k} U_s \}$$

Critère 4 : celle pour laquelle la somme des degrés des colonnes agissantes est maximum dans  $S_k$ .

$$i \text{ telle que } \sum_{j \in J_i} d^{(k)}_j = \max_{t \in I_k} \left\{ \sum_{j \in J_t} d^{(k)}_j \right\}$$

Critère 5 : celle mettant en cause le maximum de lignes différentes

$$i \text{ telle que } (|U_s| / J_s \cap J_i \neq \emptyset) = \max_{t \in I_k} \{ |U_s| / J_s \cap J_t \neq \emptyset \}$$

Critère 6 : celle qui laissera pour l'étape suivante la ligne de plus faible degré

$$i \text{ telle que } \min_l (d_{l,i}^{(k+1)}) = \min_{t \in I_k} (d_{l,t}^{(k+1)}) / l \in L_{k+1}^c$$

#### Résultats numériques :

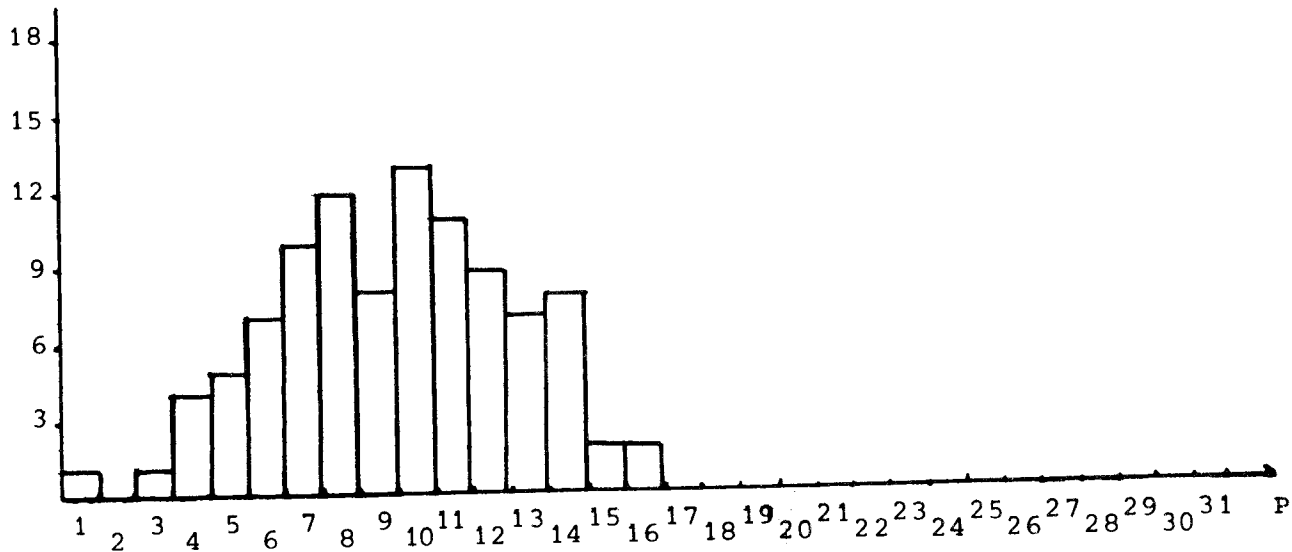
Tout ou partie des critères énumérés ci-dessus a été testé sur différentes matrices.

#### 1. Matrices tirées au hasard - critère 1

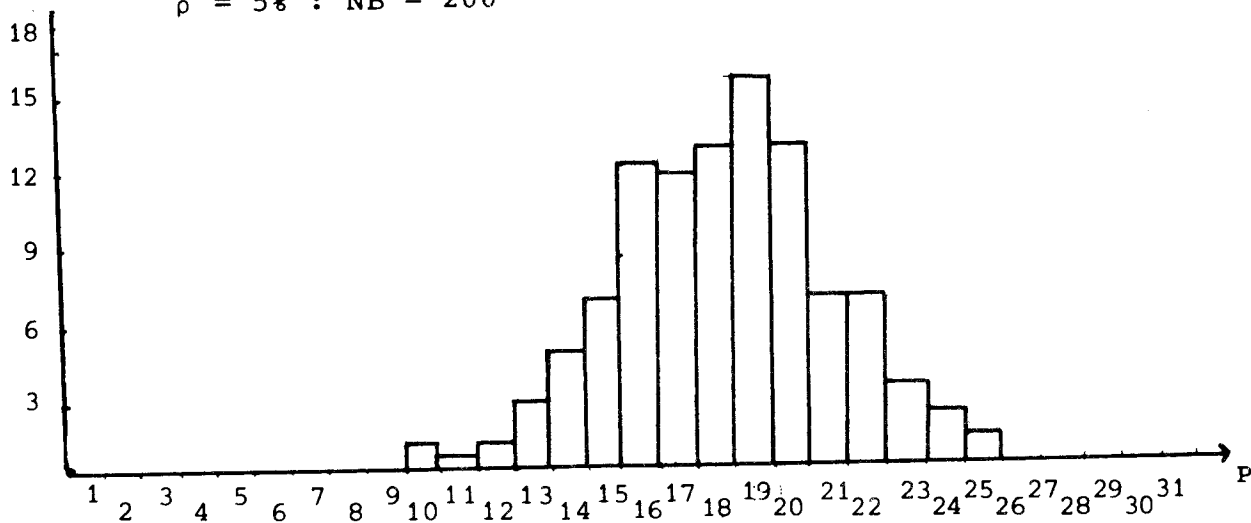
Ce sont des matrices carrées de dimension 100 dans lesquelles chaque élément a une probabilité ( $\rho$ ) d'être non nul.

Pour  $\rho$  fixé, on a obtenu une certaine distribution de la variable "nombre de paramètres" à partir d'un nombre NB de matrices

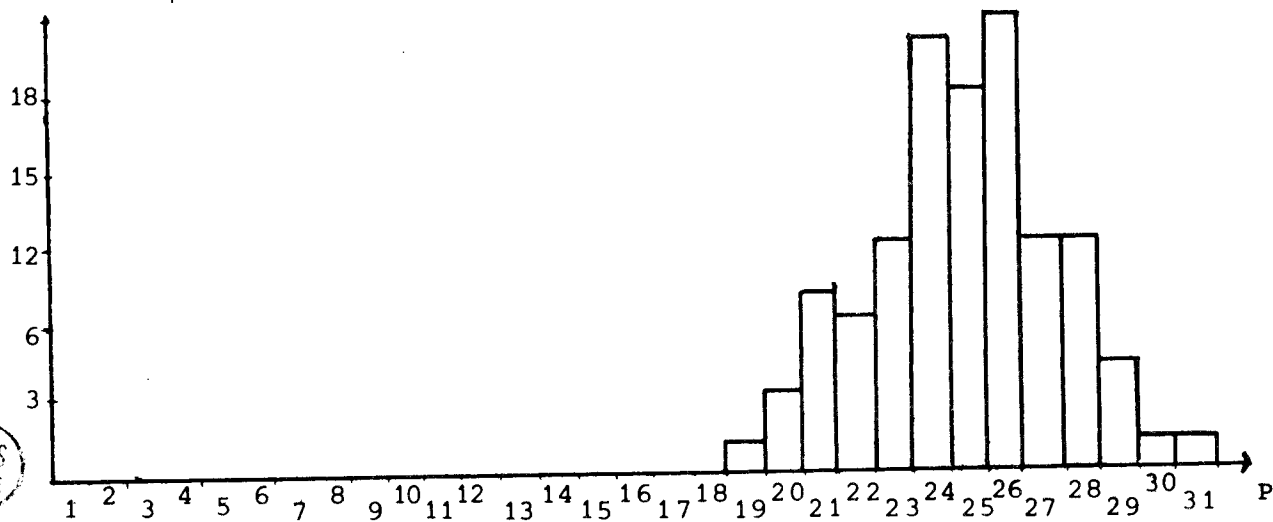
$\rho = 4\% : NB = 100$



$\rho = 5\% : NB = 200$



$\rho = 6\% : NB = 100$





On a extrait successivement de la matrice générale 12 matrices de dimension 20x20 sur laquelle les 6 critères ont été testés ; ces matrices comprenaient les colonnes :

numéros	colonnes phases	colonnes tension	colonnes puissances actives	colonnes puissances réactives
	11 à 19	1 à 10	20 à 43	44 à 50
1	11 à 19	1 à 10	20	
2	"	1 à 9	20,24	
3	"	1 à 8	20,24,26	
4	"	1 à 7	20,24,26,30	
5	"	1 à 6	20,24,26,30,35	
6	"	1 à 5	20,24,26,30,35,38	
7	"	1,3,4,5	20,24,26,30,35,38,40	
8	"	1,3,4	20,24,26,30,35,38,40	44
9	"	1,4	"	44,45
10	"	4	"	44,45,46
11	"		"	44,45,46,47
12	"	3,4,5,7	20,24	45,47,48,49,50

Résultats : le tableau suivant donne le nombre maximum de paramètres obtenus pour les différents matrices et critères :

N°	$\rho\%$		C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>	C <sub>6</sub>
1	35,25	A	5	6	6	6	6	6
		A <sup>T</sup>	7	5	7	7	7	5
2	34,5	A	4	4	6	6	4	4
		A <sup>T</sup>	5	5	5	5	5	5
3	33,75	A	4	4	4	4	4	4
		A <sup>T</sup>	4	4	4	4	4	4
4	32,5	A	4	4	4	4	4	4
		A <sup>T</sup>	4	3	3	3	3	3
5	31,25	A	5	4	4	4	4	4
		A <sup>T</sup>	3	3	3	3	3	3
6	30	A	4	4	4	4	4	4
		A <sup>T</sup>	3	3	3	3	3	3
7	28,75	A	4	4	4	4	4	4
		A <sup>T</sup>	4	4	4	4	4	4

N°	p%		C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>	C <sub>6</sub>
8	27	A	3	3	3	3	3	3
		A <sup>T</sup>	3	3	3	3	3	3
9	25,25	A	2	2	2	2	2	2
		A <sup>T</sup>	2	2	2	2	2	2
10	23	A	2	2	2	2	2	2
		A <sup>T</sup>	3	2	2	2	2	2
11	19,75	A	1	2	2	1	1	1
		A <sup>T</sup>	2	2	2	2	2	2
12	27,75	A	4	6	6	4	6	4
		A <sup>T</sup>	4	5	5	5	5	4

tableau 1

### 3) Matrices de type dispatching

Cinq matrices de type particulier ont été construites de la manière suivante :

1°) Détermination d'une matrice B (50,50)

2°) Construction de A à partir de B :

$$A = \begin{array}{|c|c|c|} \hline 1 & & \\ \hline 0 & B^{(49)} & B \\ \hline | & & \\ \hline 0 & & \\ \hline | & & \\ \hline 0 & B^{(49)} & B \\ \hline | & & \\ \hline \end{array}$$

B<sup>(49)</sup> signifiant les 49 premières colonnes de B

Ces matrices sont de même type que les matrices de base extraites de la matrice de dispatching du réseau grec (44 noeuds) Annexe 1- et peuvent symboliser des matrices de base extraites de la matrice d'un réseau électrique (50 noeuds).



On a obtenu les résultats suivants pour le nombre de paramètres sur les différentes matrices A (100,100) :

critères Aspect de B	critères						
	$\rho\%$	1	2	3	4	5	6
	5,97	8	7	10	6	6	7
	6,49	9	11	13	8	8	11
	5,93	9	7	9	7	7	7
	5,95	7	5	7	7	7	5
	5,95	8	8	10	8	8	8

tableau 2

un exemple des différentes matrices A et du meilleur résultat obtenu pour les différents critères est donné en annexe 1 - (les éléments non nuls sont symbolisés par la lettre A -)

De même, en utilisant le critère 1 et en faisant varier  $\rho$ , on obtient :

Aspect de B	$\rho\%$		9,49		9,15		8,93		8,85		7,35		6,49		5,97		5,95		5,93	
	A	$A^t$	A	$A^t$	A	$A^t$	A	$A^t$	A	$A^t$	A	$A^t$	A	$A^t$	A	$A^t$	A	$A^t$	A	$A^t$
			21	18											8	7				
							17	17	13	12									9	10
	15	16																	7	7
					22	20											8	8		

tableau 3



Enfin, si on note  $A_m$  la matrice suivante :

$$A_m = \begin{pmatrix} & C^T & B \\ 0 & B^{(49)} & B \\ 0 & & \end{pmatrix}$$

où  $C^T$  est la transposée de la matrice suivante

$$\begin{pmatrix} 1 & \\ 0 & B^{(49)} \end{pmatrix}$$

on obtient, toujours avec le critère 1, :

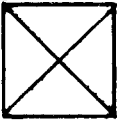


Aspect de B	$\rho$ %	A	$A^t$	$A_m$
	9,49	7	7	14
	5,95	15	16	25
	5,95	8	8	22

tableau 4

4. Remarques et commentaires

Les tableaux suivants obtenus à partir des tableaux 1 et 2 donnent le nombre de fois où chaque critère a permis d'aboutir :

- au meilleur résultat (tableau 5)
- au moins bon résultat -les 15 cas où les six critères donnent le même résultat ne sont pas comptabilisés- (tableau 6)

	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$	$C_6$
Nombres sur 29	21	23	18	24	25	26

tableau 5

	C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>	C <sub>6</sub>
Nombres sur 14	5	4	11	5	5	1

tableau 6

Remarque 1 : Aucun critère ne donne toujours le meilleur résultat.

Remarque 2 : Le critère 3 semble être le moins bien adapté, car s'il donne dix huit fois le meilleur résultat, il donne onze fois le plus mauvais.

Remarque 3 : Les différences entre le critère 1 et le critère 2 ne peuvent qu'être dues au hasard.

Remarque 4 : Le critère 6 est le plus adapté, résultat logique car il est basé sur une simulation du résultat à l'étape suivante.

Remarque 5 : Les critères 4 et 5 apparaissent comme des critères intermédiaires entre le groupe (critère 1, critère 2) et le critère 6.

on peut établir les classements suivants

1- par ordre d'intérêt croissant

3, 1 et 2, 4 et 5, 6

2- par rapport au coût croissant de mise en oeuvre

1 et 2, 3, 4, 5, 6

Aussi est-ce le critère 1 qui a toujours été retenu pour la suite des expériences numériques.

Remarque 6 : Les trois histogrammes et le tableau 3 confirment que le nombre de paramètres augmente rapidement lorsque le creux diminue. Sur les échantillons ayant permis de déterminer les histogrammes, on a obtenu en moyenne :

9,49	paramètres pour $\rho$	=	4%
18,14	"	"	= 5%
24,88	"	"	= 6%

Remarque 7 : Les résultats obtenus sur les matrices tirées au hasard sont très différents de ceux obtenus sur les matrices de type dispatching avec sensiblement le même creux (94%) : 24,88 paramètres en moyenne pour les premières contre 8 pour les secondes.

Cette différence s'explique par la construction des matrices de type dispatching : à chaque ligne de la matrice correspond une ligne ayant la même structure (ie les deux lignes ont leurs éléments non nuls dans les mêmes colonnes) et surtout, les matrices B ont une forme particulière, ce qui favorise la similitude des structures de groupes de lignes, donc permet d'arriver très vite dans la procédure de résolution à l'"élimination" de paramètres.

Ce fait est confirmé par le tableau 4 qui donne les résultats obtenus sur trois des matrices de type dispatching lorsqu'elles perdent leur structure particulière.

#### B-3-6 Cas défavorable [23]

Ce sont les matrices de configuration ou plans projectifs finis, [25], telles que :

- $|N_i \cap N_j| = 1 \quad \forall i, \forall j, i \neq j$  à k éléments par ligne et par colonne.
- $k = m + 1$  où m premier ou puissance de premier
- $n = m^2 + m + 1$

Exemple :  $m = 3 \quad k = 4 \quad n = 13$

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	x	x	x	x									
2	x				x	x	x						
3		x			x			x	x				
4			x			x		x		x			
5			x				x		x		x		
6				x	x					x	x		
7				x		x			x			x	
8		x					x			x		x	
9	x							x			x	x	
10				x			x	x					x
11	x								x	x			x
12		x				x				x			x
13			x		x							x	x

on obtient  $p \geq \frac{n-1}{2} = \frac{m(m+1)}{2}$

dans l'exemple précédent,  $p = 6$

Démonstration :

Quelle que soit la première équation, elle entraînera l'introduction de  $(k-1)$  paramètres.

Dans le meilleur des cas, la seconde équation amènera  $(k-2)$  nouveaux paramètres puisque  $|N_i \cap N_j| = 1 \forall i, j, i \neq j$ . De même, la troisième équation amènera  $(k-3)$  nouveaux paramètres

|  
|  
|

la  $(k-1)^{\text{e}}$  équation amènera 1 paramètre  
donc  $p \geq (k-1) + (k-2) + \dots + 1$

$$\sum_{i=1}^{(k-1)} i = \frac{k(k-1)}{2} = \frac{m(m+1)}{2} = \frac{(n-1)}{2}$$

$$p \geq \frac{(n-1)}{2}$$

Ce cas est, à notre connaissance, le plus défavorable que l'on puisse trouver à dimension de matrice et à densité égales.

### B-4- Phase 2 : résolution

La première phase ayant permis de déterminer la permutation  $f$  et la dimension  $p$  du tableau ( $n \times p$ ) nécessaire à la résolution, celle-ci est alors possible. Durant la résolution, deux éléments caractéristiques de la méthode apparaissent :

- le choix du pivot
- l'élimination de paramètres.

#### B-4-1 Le choix du pivot

On a vu, sur les différents exemples donnés dans ce chapitre, que de chaque équation était tirée l'expression d'une variable.

Si l'expression est réduite à un nombre, c'est la valeur finale de la variable, sinon l'expression est une fonction de variables "paramètres".

Le coefficient de cette variable est donc le terme utilisé comme pivot.

Deux cas se présentent à l'étape  $k$  :

- on exprime une "nouvelle" variable, ce qui signifie que  $A_{ik} = 0$  pour  $i < k$ .

Le pivot est alors un terme de la matrice initiale, ce qui est important car,

- \* on ne commet aucune erreur sur sa valeur exacte
- \* dans la mesure où tous les coefficients de la matrice initiale sont du même ordre de grandeur, il n'est pas la cause de grandes erreurs de calcul et son choix se rapproche de celui donné par le critère de Durand (voir page 3-chapitre I)

De toute manière, on peut toujours rechercher le "meilleur" pivot dans  $J_i^{(k)} = \{j \in C_k^C / A_{ij} \neq 0\}$ , dans la mesure où cet ensemble n'est pas réduit à 1 seul élément.

- on exprime une variable paramètre en fonction des autres paramètres (ie on "élimine" un paramètre).

C'est dans ce cas qu'une recherche sur le "meilleur" pivot s'impose d'où l'option "pivotage partiel sur les colonnes".

Il faut remarquer que pour les matrices creuses, le premier cas se rencontre généralement beaucoup plus que le second.

#### B-4-2 l'Elimination d'un paramètre

Ce qu'il faut surtout remarquer dans ce cas, c'est qu'on "élimine" un paramètre dès que possible. De même, on recalcule aussitôt l'expression des variables déjà exprimées en fonction de ce paramètre.

Cela permet qu'à une étape donnée dans le tableau (nxp) nécessaire à la résolution, une colonne soit indiquée comme le paramètre auquel elle se rapporte.

Ainsi, cette colonne peut être plusieurs fois indiquée au cours de la résolution.

Il suffit donc de deux tableaux, respectivement de dimension (n) et (p), pour indiquer tous les éléments qui pourront se trouver en cours de résolution dans le tableau auxiliaire .

Enfin, l'expression d'une variable à l'étape k tenant compte des calculs effectués aux étapes 1 à (k-1), c'est une expression de substitution "minimale", qu'on utilise, par rapport à la fonction de permutation f.

exemple :

A =			X <sub>1</sub>
k	/		X <sub>2</sub>

à l'étape k, la valeur de la variable  $x_1$  aura été déterminée, et c'est cette valeur qu'on utilisera dans la suite de la résolution.

### B-5- La RCL factorisation

Le principe présenté dans le paragraphe A7 de ce chapitre reste le même, seules quelques remarques viennent le compléter dans le cas des matrices creuses.

(k)  
1- La matrice (C) ne sera pas réduite à une matrice diagonale que si on élimine un paramètre à l'étape k. L'autre finalité possible à l'étape k étant l'expression d'une nouvelle variable en fonction des paramètres déterminés précédemment, elle correspond au cas où  $(A_{ik} = 0 \quad i < k)$  - on aura donc  $(L^{(k)} \quad L^{(2)} \quad L^{(1)} \quad C \quad A)_{ik} = 0$  pour  $i < k$  donc (C) est une matrice diagonale.

(k)  
2- Les éléments non triviaux composant les matrices C sont ceux apparaissant dans la colonne du tableau (nxp) correspondant au paramètre qu'on élimine à l'étape k (à une division près par le pivot).

exemple :

Soit la matrice A

	1	2	3	4	5	6
1	x		x			x
2		x	x			x
3		x	x			x
4	x			x		x
5		x		x	x	x
6			x	x	x	x

à l'étape 3, le tableau (nxp) se transforme de la manière suivante, si on élimine le paramètre d'indice 3 :

	3	6
1	t <sub>1</sub>	v <sub>1</sub>
2	t <sub>2</sub>	v <sub>2</sub>
?	t <sub>3</sub>	v <sub>3</sub>

	6
1	v' <sub>1</sub>
2	v' <sub>2</sub>
3	v' <sub>3</sub>

alors

$$\begin{matrix}
 (3) \\
 C
 \end{matrix}
 =
 \begin{pmatrix}
 1 & -t_1/t_3 & & & \\
 & 1 & -t_2/t_3 & & \\
 & & 1/t_3 & & \\
 & & & 1 & \\
 & & & & 1 & \\
 & & & & & 1
 \end{pmatrix}$$

Dans cet exemple, les matrices  $C^{(1)}$ ,  $C^{(2)}$ ,  $C^{(4)}$  et  $C^{(5)}$  seront des matrices diagonales avec un élément différent de 1 qui sera respectivement

$$1/a_{11}, 1/a_{22}, 1/a_{44} \text{ et } 1/a_{55}$$

On retrouve là le fait que le pivot est un terme d'origine de la matrice A développé en B-4-1 chapitre II.

3- Enfin, il faut remarquer que dans la mesure où les matrices A et  $A^{-1}$  doivent être mémorisées simultanément, la factorisation RCL peut être d'un emploi avantageux du point de vue encombrement mémoire car seules les matrices  $C^{(k)}$  (non triviales) sont à mémoriser



## C/ ADAPTATION À QUELQUES PROBLÈMES PARTICULIERS

La méthode générale qui vient d'être développée s'adapte très facilement à certains problèmes souvent rencontrés dans la littérature.

1) Résolution d'une suite de systèmes  $A^{(k)} x^{(k)} = b^{(k)}$

-  $A^{(k)} = A \forall k$

on conservera  $A^{-1}$  sous la forme RCL

- les matrices  $A^{(k)}$  ont la même structure.

Il suffit de rechercher une seule fois la permutation  $f$  et de résoudre les différents systèmes d'après celle-ci.

2) Systèmes à matrice particulière

- Matrice symétrique pleine.

Soit  $A$  la matrice d'un système de dimensions  $(n \times n)$ , symétrique et pleine.

Alors  $(n^2+n)$  mémoires suffisent à mémoriser  $A$  et  $A^{-1}$  simultanément par la factorisation RCL.

- Matrice blocs diagonaux.

$$A = \begin{array}{|c|c|c|} \hline \blacksquare & & \\ \hline & \blacksquare & \\ \hline & & \blacksquare \\ \hline \end{array}$$

soient  $N_1, N_2, \dots, N_k$  les sous-ensembles de  $N$  déterminant les blocs diagonaux et  $P_1, P_2, \dots, P_k$  le nombre maximum de paramètres dans chaque bloc. On note  $n_1, \dots, n_k$  les cardinaux de  $N_1, \dots, N_k$  alors il suffira de prendre un tableau de dimension

$(\sup_{1 \leq i \leq k} (n_i) \times \sup_{1 \leq i \leq k} (p_i))$  pour pouvoir résoudre le système.

## CHAPITRE III

### APPLICATION À LA MÉTHODE SIMPLICIALE

## INTRODUCTION

---

La méthode des paramètres exposée au chapitre précédent trouve une application idéale dans la méthode simpliciale.

C'est cette méthode ainsi que ses différentes mises en application qui sont rappelées dans la section A.

De même, l'adaptation de la méthode des paramètres à la méthode simpliciale est présentée dans la section B.

Les résultats des expériences numériques effectuées dans ce cadre sur différents problèmes sont donnés dans la section C.

## A/ LA MÉTHODE SIMPLICIALE ET SES DIFFÉRENTES MISES EN APPLICATION

### A-1- Rappel de la méthode simpliciale [26]

soit le programme linéaire  
maximiser  $f \cdot x$

$$Ax \leq b$$

$$x \geq 0$$

où  $A$  est une matrice  $m \times n$ , indicée par  $L \times J$ ,  $x \in \mathbb{R}^n$ ,  $f^t \in \mathbb{R}^n$ ,  $b \in \mathbb{R}^m$

Définitions :

\*  $I$  est une base  $\Leftrightarrow \begin{cases} I \in J \\ |I| = |L| \\ A^I \text{ est inversible} \end{cases}$

une solution de base réalisable  $x(I)$  est telle que  $\begin{cases} A^I x_I = a \\ x_I \geq 0 \\ x_{\bar{I}} = 0 \end{cases}$

\*  $d(I) = f - f^I (A^I)^{-1} A$  est le vecteur critère de candidature

Théorème : Si pour une base  $I$  réalisable on a  $d(I) \leq 0$ ,  
alors  $x(I)$  est solution optimale du problème.

Principe de la méthode :

La méthode simpliciale définit une suite de valeurs  $f \cdot x(I)$  monotone non décroissante, en se déplaçant d'une base  $I$  à une base voisine  $I' = I + r - s$ ,  $r$  et  $s$  étant définis comme suit :

1)  $d(I) \not\leq 0$ , alors on choisit  $s \in \bar{I}$  tel que  $d^s(I) > 0$

2) l'indice  $r$  quittant la base est alors sélectionné de la façon suivante :

$$r : \frac{t_r(I)}{T_r^s(I)} = \min \left\{ \frac{t_i(I)}{T_i^s(I)} \mid T_i^s(I) > 0, i \in I \right\}$$

$$\text{où } t(I) = (A^I)^{-1} a$$

$$T(I) = (A^I)^{-1} A$$

Calculs à effectuer :

1) pour la détermination de  $s$ , il faut calculer  $d(I)$

$$d(I) = f - f^I (A^I)^{-1} A$$

soit  $d(I) = f - uA$  avec  $uA^I = f^I$

2) pour la détermination de  $r$ ,  $t(I)$  et  $T^S(I)$  sont nécessaires.

Il faut donc, à chaque changement de base, résoudre trois systèmes linéaires :

$$uA^I = f^I$$

$$A^I t = a$$

$$A^I T^S = A^S$$

c'est sur la manière de résoudre ces trois systèmes que diffèrent les mises en application de la méthode présentées ci-dessous.

A-2- Différentes mises en application

1 - Calculs directs

On résoud directement les trois systèmes à chaque itération et seules les données du problème doivent être mémorisées.

Ce principe n'est en fait acceptable que pour des petits programmes linéaires ou des programmes linéaires de structures particulières, (problèmes de transport par exemple), car sinon la résolution des trois systèmes est longue et coûteuse.

2 - Méthode de l'inverse explicite

On conserve en mémoire la matrice  $A$  et l'inverse de la matrice de base  $A^I$ , mise à jour à chaque itération.

En effet, deux matrices de base  $A^I$  et  $A^{I'}$  successives diffèrent d'une seule colonne. On peut donc obtenir la matrice inverse de  $A^{I'}$  en multipliant l'inverse de  $A^I$  par une matrice d'élimination de Jordan  $E_I$  telle que :

$$E_I = \begin{array}{|c|c|c|} \hline \begin{array}{c} 1 \\ \diagdown \\ 1 \end{array} & \begin{array}{c} -T_1^S/T_r^S \\ -T_{r-1}^S/T_r^S \end{array} & 0 \\ \hline & 1/T_r^S & \\ \hline 0 & \begin{array}{c} -T_{r+1}^S/T_r^S \\ -T_n^S/T_r^S \end{array} & \begin{array}{c} 1 \\ \diagdown \\ 1 \end{array} \\ \hline \end{array}$$

où  $T^S$  est la solution du système  $A^I T^S = A^S$ , on obtient  
 $(A^{I'})^{-1} = E_I (A^I)^{-1}$ .

Il faut en pratique disposer de  $m^2$  mémoires pour mémoriser l'inverse; même si  $A^I$  est une matrice creuse,  $(A^I)^{-1}$  ne l'est plus après un certain nombre d'itérations.

### 3- La factorisation PFI (product form of inverse).

s'inspire du principe exposé ci-dessus, où à l'itération  $k$  :

$$\begin{aligned} (A^{I_k})^{-1} &= E_{k-1} (A^{I_{k-1}})^{-1} \text{ soit} \\ (A^{I_k})^{-1} &= E_{k-1} E_{k-2} \dots E_2 E_1 (A^{I_1})^{-1} \end{aligned}$$

on mémorise l'inverse de la matrice de base à l'itération  $k$  sous forme d'un produit de matrices élémentaires  $E_l$ ,  $1 \leq l \leq k-1$  et, si nécessaire, de la matrice  $(A^{I_1})^{-1}$ .

### 4 - Factorisation EFI (effective form of inverse)

Cette méthode est basée sur le principe de la décomposition LU d'une matrice inspirée de la méthode de Gauss

$$\begin{aligned} A^I &= LU \text{ avec } L \text{ triangulaire inférieure} \\ &\quad U \text{ triangulaire supérieure} \end{aligned}$$

#### Actualisation de la décomposition

soient  $A^I = \{A_1, A_2, \dots, A_r, \dots, A_m\}$  où  $A_r$  est la colonne sortant de la base et  $A^{I'} = \{A_1, A_2, \dots, A_s, \dots, A_m\}$  où  $A_s$  est la colonne entrant dans la base

$$\text{on a } A^I = L U$$

$$\text{soit } H^{I'} = (L)^{-1} (A^{I'})$$

$$= \{(L)^{-1} A_1, (L)^{-1} A_2, \dots, (L)^{-1} A_s, \dots, (L)^{-1} A_m\}$$

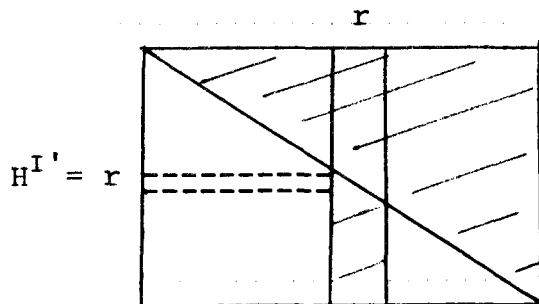
mais  $(L)^{-1}(A^I) = \{U_1, U_2, \dots, U_r, \dots, U_m\} = U$

matrice triangulaire supérieure

on en déduit :

$$H^{I'} = \{U_1, \dots, U_{r-1}, (L)^{-1}A_S, U_{r+1}, \dots, U_m\}$$

soit une matrice de la forme



pour transformer  $H^{I'}$  en  $U$  triangulaire supérieure, il suffit de triangulariser la partie inférieure de  $H^{I'}$  à partir des lignes et colonnes d'indice  $r$ .

On peut ainsi obtenir une meilleure précision que dans les méthodes 2 et 3, où le pivot  $T_r^S$  est imposé, car on a la possibilité de choisir les différents pivots utilisés lors de la triangularisation.

### variantes

Outre la méthode présentée ci-dessus, l'actualisation a donné lieu à plusieurs variantes, en particulier

- l'actualisation LU modifiée de Bartels et Golub
- l'actualisation LU de Tomlin

on pourra consulter à ce sujet la référence [27],

Ces différentes méthodes (sauf la première) sont d'un emploi très général et ne nécessitent aucune hypothèse sur la structure de la matrice du programme linéaire.

Il va de soi qu'elles ont pu être adaptées dans le cas de PL dont la matrice présente une structure particulière.

## B/ CAS DE LA MÉTHODE DES PARAMÈTRES

L'adaptation de la méthode des paramètres à la méthode simpliciale peut se faire de deux manières :

- 1- en utilisant la factorisation RCL
- 2- avec résolution directe à chaque itération.

La première ne nécessite aucune hypothèse particulière sur la structure de la matrice du programme linéaire et peut être utilisée en tenant compte de la propriété

$$l_{kj}^{(k)} = a_{kj}^{(2k-1)} = -a_{kj}^{(1)} \quad \begin{array}{l} 1 \leq j \leq k-1 \\ k = 2, \dots, n \end{array}$$

(Chapitre II - A - 7)

Il suffit donc de mémoriser les matrices  $C^{(k)}$ ,  $k = 1, n$ , de les mettre à jour à chaque itération et de savoir repérer les matrices  $L^{(k)}$  sur les données de la matrice A.

Ce procédé entre dans le même cadre que les méthodes 3 et 4 exposées dans la section A mais ne tient pas compte d'une caractéristique généralement rencontrée dans les programmes linéaires de grande taille.

La matrice du PL est très creuse, creux qu'on retrouve dans les matrices de base  $A^I$ .

C'est sur cette caractéristique que repose la seconde manière d'adapter la méthode des paramètres à la méthode simpliciale et qu'on étudie ici.

### B-1- Résolution directe des systèmes linéaires à chaque itération.

Pour tenir compte de la remarque contenue en A-2-1 de ce chapitre, l'adaptation présentée ici est celle avec calculs directs à chaque itération sur des PL dont la caractéristique principale de la matrice n'est autre que d'être creuse.



Ce principe présente plusieurs avantages :

1 - Il n'y a pas cumul des erreurs de calcul

En effet, les solutions des différents systèmes sont calculées à chaque itération à partir de la matrice  $A^I$  et non pas à partir de l'inverse  $(A^I)^{-1}$  obtenue par les mises à jour successives aux itérations précédentes.

2 - Pour démarrer l'algorithme, il suffit de connaître une base réalisable  $I$  quelconque.

L'inverse  $(A^I)^{-1}$  n'étant pas nécessaire, on peut démarrer l'algorithme avec n'importe quelle matrice  $(A^I)$  pourvu que  $I$  soit une base réalisable.

3 - Tenant compte du creux de  $(A^I)$  à chaque itération, tous les éléments nécessaires à la résolution se trouvent en mémoire centrale d'ordinateur, ce qui permet une programmation relativement simple de la méthode et la possibilité de travailler sans fichiers intermédiaires.

4 - Pour la même raison que précédemment, la taille des PL que l'on peut résoudre en mémoire centrale d'ordinateur s'en trouve considérablement augmentée.

#### B-2- Remarques concernant l'adaptation

1- Pour une matrice donnée  $(A^I)$  et lorsque les permutations  $f$  et  $g$  (B-3-3 chapitre II) sont déterminées, alors les matrices  $(P A Q)$  et  $(Q' A^t P')$  présentent le même nombre maximum de paramètres.

$P$  et  $Q$  sont les matrices de permutation des fonctions  $f$  et  $g$ .

$P'$  et  $Q'$  sont les matrices de permutation des fonctions  $f'$  et  $g'$

telles que :

$$f : \{1, 2, \dots, n\} \rightarrow \{f(1), f(2), \dots, f(n)\}$$

$$g : \{1, 2, \dots, n\} \rightarrow \{g(1), g(2), \dots, g(n)\}$$

$$f'(k) = n - f(k) + 1$$

$$g'(k) = n - g(k) + 1$$

Démonstration :

pour  $(P A Q)$ ,  $a_{ij} = 0$  pour  $1 \leq i \leq n-p-1$  et  $i+p+1 \leq j \leq n$

pour  $(P A Q)^t$ ,  $a_{ij} = 0$  pour  $1 \leq j \leq n-p-1$  et  $j+p+1 \leq i \leq n$

par définition des fonctions  $f'$  et  $g'$ , on obtient :

$$j \rightsquigarrow n-j+1$$

$$i \rightsquigarrow n-i+1$$

donc, pour  $(Q' A^t P')$ ,  $a_{ij} = 0$  pour  $1 \leq n-j+1 \leq n-p-1$   
 $j+p+1 \leq n-i+1 \leq n$

$$\text{soit } p+2 \leq j \leq n$$

$$\text{et } 1 \leq i \leq j-p-1$$

ou ce qui est équivalent  $1 \leq i \leq n-p-1$  et  $i+p+1 \leq j \leq n$

Il suffira donc de déterminer une seule fois  $f$  et  $g$  pour pouvoir résoudre les trois systèmes.

2- Lors d'un changement de base de  $I$  à  $I'$ , si on note  $P$  et  $Q$  les matrices de permutation des fonctions  $f$  et  $g$  relatives à  $A^I$  alors le nombre maximum de paramètres de la matrice  $(P A^{I'} Q)$  est  $p'$  tel que  $p-1 \leq p' \leq p+1$

Démonstration :

D'après la définition du problème, et pour une permutation  $f$  :

$$P = \max \{ (|N_{f(i)}^I| - f_{(i)}) / i \in N \}$$

(B-3-3 chapitre II)

les matrices  $A^I$  et  $A^{I'}$  ne diffèrent que d'une seule colonne, on obtient que les  $N_{f(i)}$  pour la matrice  $I'$  sont telles que :

$$|N_{f(i)}^I| - 1 \leq |N_{f(i)}^{I'}| \leq |N_{f(i)}^I| + 1 \quad \forall i \in N$$

on a

$$P' = \max \{ (|N_{f(i)}^{I'}| - f_{(i)}) / i \in N \}$$

donc  $P-1 \leq P' \leq P+1$

Cette remarque peut donner lieu à une variante de l'adaptation de la méthode des paramètres à la méthode simpliciale ; en effet, le tableau auxiliaire  $(n \times p)$  nécessaire à la résolution d'un système devant être dimensionné au début du programme, on pourra ne déterminer les permutations  $f$  et  $g$  qu'aux itérations où le nombre de paramètres rencontrés en cours de résolution risque

de dépasser la valeur donnée dans le dimensionnement de ce tableau.

### 3 - Cas de la méthode simpliciale en variables bornées.

On peut introduire les contraintes de borne directement dans la matrice  $A$  du PL à résoudre, ce qui a pour effet d'augmenter la dimension des systèmes à résoudre mais non le nombre de paramètres de ces systèmes.

On ne traite que le cas où  $x_i \leq x_m$  (le cas  $x_i \geq x_m$  donne le même résultat).

Deux possibilités pour une matrice de base  $A^I$ .

.  $x_i$  est de base et à sa borne supérieure : la variable est entièrement déterminée et n'interviendra pas dans les calculs variables-paramètres de la résolution

.  $x_i$  et la variable d'écart associée à la contrainte  $y_i$  sont de base.

Il existe donc dans la matrice  $(A^I)$  une colonne de degré 1, celle relative à  $y_i$ ) et on peut se ramener à la forme donnée en B.3.1, chapitre III.

## C/ EXPÉRIENCES ET RÉSULTATS NUMÉRIQUES

Les expériences numériques décrites ci-dessous ont un double but :

- comparer la méthode simpliciale avec inverse explicite et celle avec résolution directe par la méthode des paramètres.
- Vérifier certaines caractéristiques de la méthode des paramètres et mettre en évidence le fait qu'elle permet de traiter des programmes linéaires de grande taille en mémoire centrale.

### C-1- : Conditions d'expériences et de validité des résultats

#### Les méthodes d'indiciage [6]

Seuls les éléments non nuls de la matrice du PL ont été mémorisés en double précision. Les méthodes d'indiciage diffèrent cependant pour les deux méthodes :

Inverse explicite : double indiciage :  
chaque élément de A est repéré par son indice de ligne et son indice de colonne, les éléments considérés colonne par colonne, étant placés dans un tableau unidimensionnel.

Méthode des paramètres : triple indiciage :  
au schéma précédent, s'ajoute un tableau unidimensionnel donnant les indices de colonne des éléments de A considérés ligne par ligne. Ce schéma d'indiciage permet de déterminer facilement  $J_i = \{j/A_{ij} \neq 0\}$  et  $J_j = \{i/A_{ij} \neq 0\}$ , éléments nécessaires pour la détermination des matrices de permutation f et g.

#### Les variables d'écart et artificielles :

Dans les deux cas, leur coefficient ( $\pm 1$ ) a été considéré comme un élément non nul de A et à ce titre, indicé comme les autres valeurs.

Les contraintes de borne sur les variables : Elles ont été prises en compte comme les contraintes initiales du problème, ce qui a pour effet d'augmenter la taille des différents tableaux

nécessaires aux résolutions. On pourra donc dans ce cas établir une comparaison sur des nombres de mémoires "corrigés" nécessaires pour la résolution.

#### Utilisation de la méthode mixte [8]

Lorsque des variables artificielles ont été introduites, c'est la méthode mixte qui a été employée pour résoudre le problème.

##### - Structure particulière de la matrice de base

Il n'a jamais été tenu compte d'une structure particulière, celle décrite en B-3-1 chapitre II par exemple, des matrices de base lors de la résolution des systèmes linéaires.

- calculs : ils ont toujours été effectués en double précision

##### - Pour l'inverse explicite :

Les opérations portant sur des éléments nuls n'ont pas été effectuées lors de la résolution des systèmes et de la mise à jour de l'inverse.

##### - Pour la méthode des paramètres :

A chaque itération, les éléments composant la matrice de base  $(A^I)^t$  ont été recopiés dans un tableau auxiliaire AU, indicé par ILU et ICU dans l'ordre donné par la fonction de permutation f. A partir de la fonction de permutation g, on peut ensuite transposer facilement  $(A^I)^t$  pour résoudre les deux derniers systèmes (les calculs ont été effectués à partir de AU).

Lors de l'élimination d'un paramètre, c'est le coefficient le plus grand en valeur absolue des coefficients des variables "paramètre" qui détermine le paramètre à éliminer (option pivotage sur les colonnes). Lorsqu'on exprime une "nouvelle" variable en fonction des paramètres, c'est la première, par ordre d'apparition, "nouvelle" variable qui est automatiquement choisie.

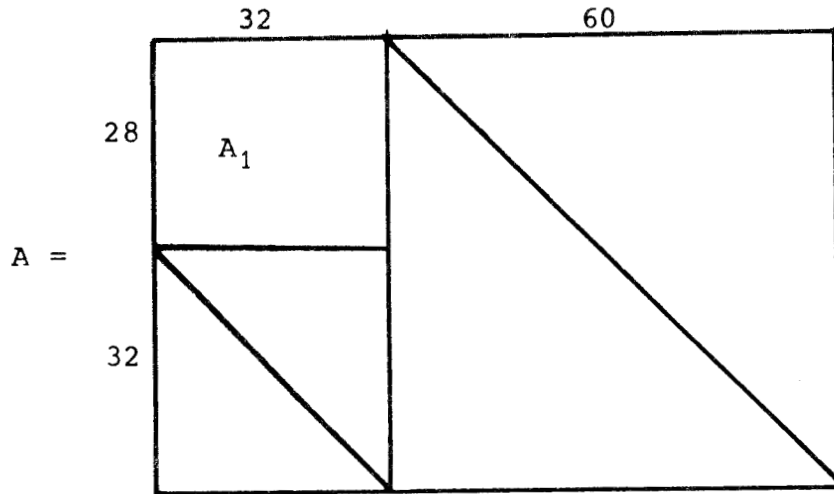
Le Code de programmation - voir annexe II pour la méthode des paramètres - langage FORTRAN -

## C-2 Résultats

- . sept problèmes ont été résolus :
  - les trois premiers ont permis de comparer deux adaptations de la méthode simpliciiale: inverse explicite et résolution directe par la méthode des paramètres ;
  - les deux suivants, deux modèles énergétiques, ont donné lieu à des programmes linéaires de grande taille qui n'ont été résolus qu'avec la méthode des paramètres.
  - les deux derniers sont des programmes linéaires rencontrés lors de la résolution d'un problème de dispatching électrique.
- . les temps sont donnés en seconde
- . l'ordinateur utilisé est un IRIS 80.

Problème n° 1 :

Problème n° 15 de [29]



à l'origine :

$$\left\{ \begin{array}{l} 32 \text{ variables} \\ 28 \text{ contraintes } \leq \\ 32 \text{ contraintes de borne } \leq \\ A_1 : 236 \text{ éléments non nuls } (\rho = 26,34\%) \end{array} \right.$$

problème à traiter :

$$\left\{ \begin{array}{l} 92 \text{ variables} \\ 60 \text{ contraintes en égalité} \\ A : 328 \text{ éléments non nuls } (\rho = 5,95\%) \end{array} \right.$$

Résultats

	Paramètres (1)	Inverse explicite(2)	Rapport (2/1)
Nombre d'itérations	40	38	
Temps moyen par itération	0,17	0,13	0,76
Nombre de mémoires utilisées	4726*	8945	1,89
pour les données **	1464	1136	
pour les calculs	3262*	7809	2,39

\* Valeur correspondant à un nombre possible de paramètres égal à 10.

\*\* Données en simple précision.

Le nombre maximum de paramètres est égal à 6.

Evolution du nombre de paramètres

Numéro de l'itération	5	10	15	20	25	30	35	40
Nombre de paramètres	0	1	3	3	3	4	5	3

Nombre de paramètres	0	1	2	3	4	5	6
Nombre d'itérations	5	5	4	13	10	2	1

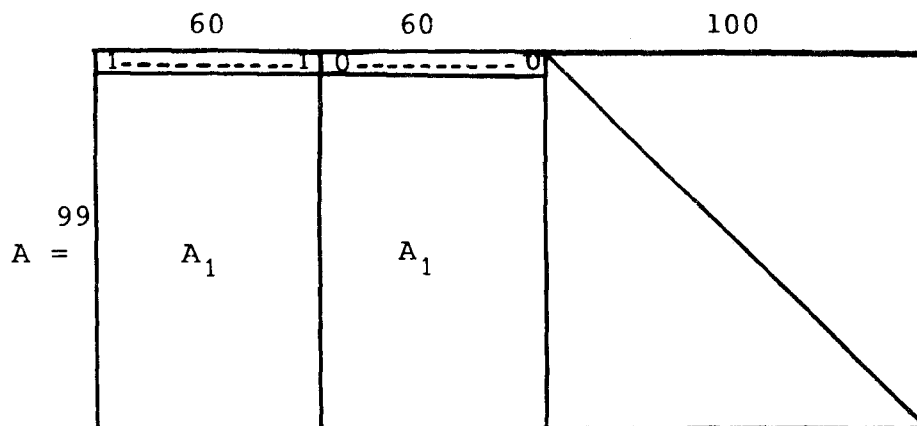
Nombre moyen de paramètres par itération : 2,7





Problème n° 2 :

Problème conçu d'après la méthode exposée dans [30]



à l'origine :

$$\left\{ \begin{array}{l} 120 \text{ variables} \\ 100 \text{ contraintes en égalité} \\ A_1 : 270 \text{ éléments non nuls } (\rho = 5\%) \end{array} \right.$$

Problème à traiter :

$$\left\{ \begin{array}{l} 220 \text{ variables} \\ 100 \text{ contraintes en égalité} \\ A : 700 \text{ éléments non nuls } (\rho = 3,18\%) \end{array} \right.$$

Résultats

	Paramètres (1)	Inverse explicite(2)	Rapport (2/1)
Nombre d'itérations	173	178	
Temps moyen par itération	0,41	0,45	1,08
Nombre de mémoires utilisées	9220 *	23733	2,57
pour les données **	3120	2420	
pour les calculs	6100 *	21313	3,49

\* Valeur correspondant à un nombre possible de paramètres égal à 10.

\*\* Données en simple précision.

Le nombre maximum de paramètres est égal à 8.

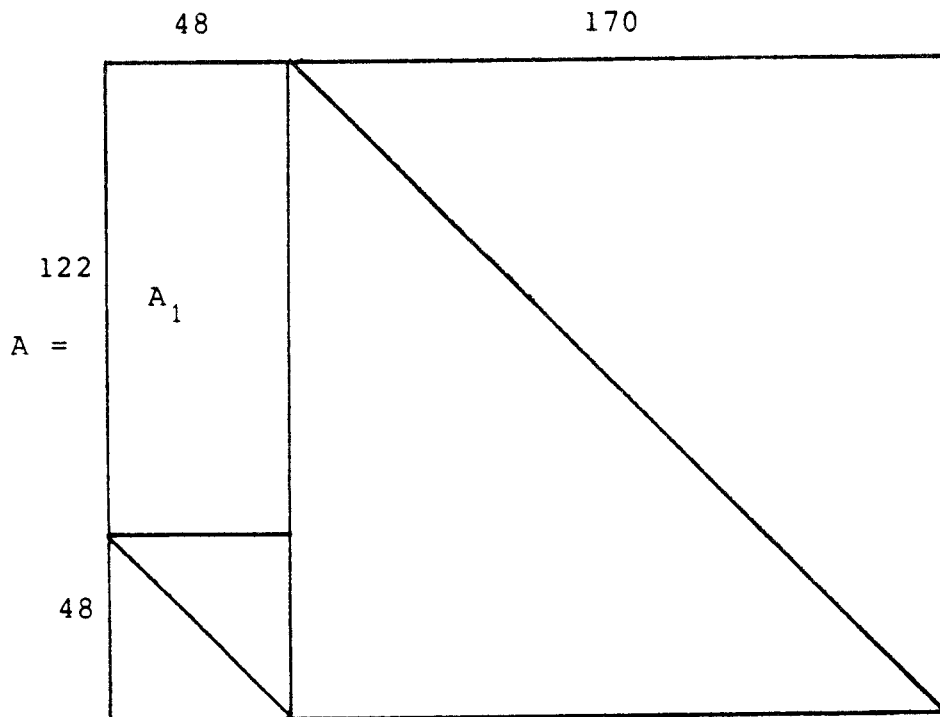
Evolution du nombre de paramètres

Numéro de l'itération	20	40	60	80	100	120	140	160	173
Nombre de paramètres	1	1	3	6	4	3	5	5	4

Nombre de paramètres	0	1	2	3	4	5	6	7	8
Nombre d'itérations	1	37	7	24	31	30	21	6	3

Nombre moyen de paramètres par itération : 3,40



Problème n° 3 :

à l'origine :

$$\left\{ \begin{array}{l} 48 \text{ variables} \\ 122 \text{ contraintes } \leq \\ 48 \text{ contraintes de borne } \leq \\ A_1 : 718 \text{ éléments non nuls } (\rho = 12,25\%) \end{array} \right.$$

Problème à traiter :

$$\left\{ \begin{array}{l} 218 \text{ variables} \\ 170 \text{ contraintes en égalité} \\ A : 936 \text{ éléments non nuls } (\rho = 2,79\%) \end{array} \right.$$

Résultats : 1°) Pour la minimisation

	Paramètres (1)	Inverse explicite(2)	Rapport (2/1)
Nombre d'itérations	102	97	
Temps moyen par itération	0,7	0,996	1,42
Nombre de mémoires utilisées	15555 *	64100	4,12
pour les données **	5520	4584	
pour les calculs	10036	59516	5,93

\* Valeur correspondant à un nombre possible de paramètres égal à 10.

\*\* Données en double précision.

Le nombre maximum de paramètres est égal à 6.

Evolution du nombre de paramètres

Numéro de l'itération	10	20	30	40	50	60	70	80	90	102
Nombre de paramètres	0	0	2	1	3	5	3	5	2	1

Nombre de paramètres	0	1	2	3	4	5	6
Nombre d'itérations	28	15	16	12	14	15	2



Nombre moyen de paramètres par itération : 2,2156.

2°) pour la maximisation :

	Paramètres (1)	<del>Inverse</del> Explicité (2)	Rapport (2/1)
Nombre d'itérations	112	120	
Temps moyen par itération	0,75	1,36	1,81

Le nombre maximum de paramètres est égal à 6.

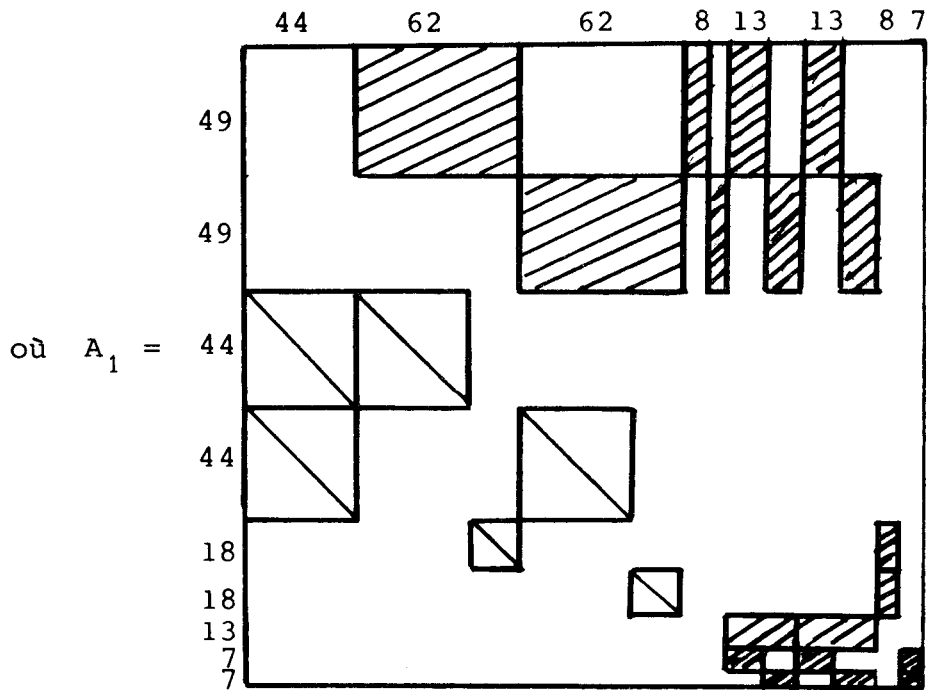
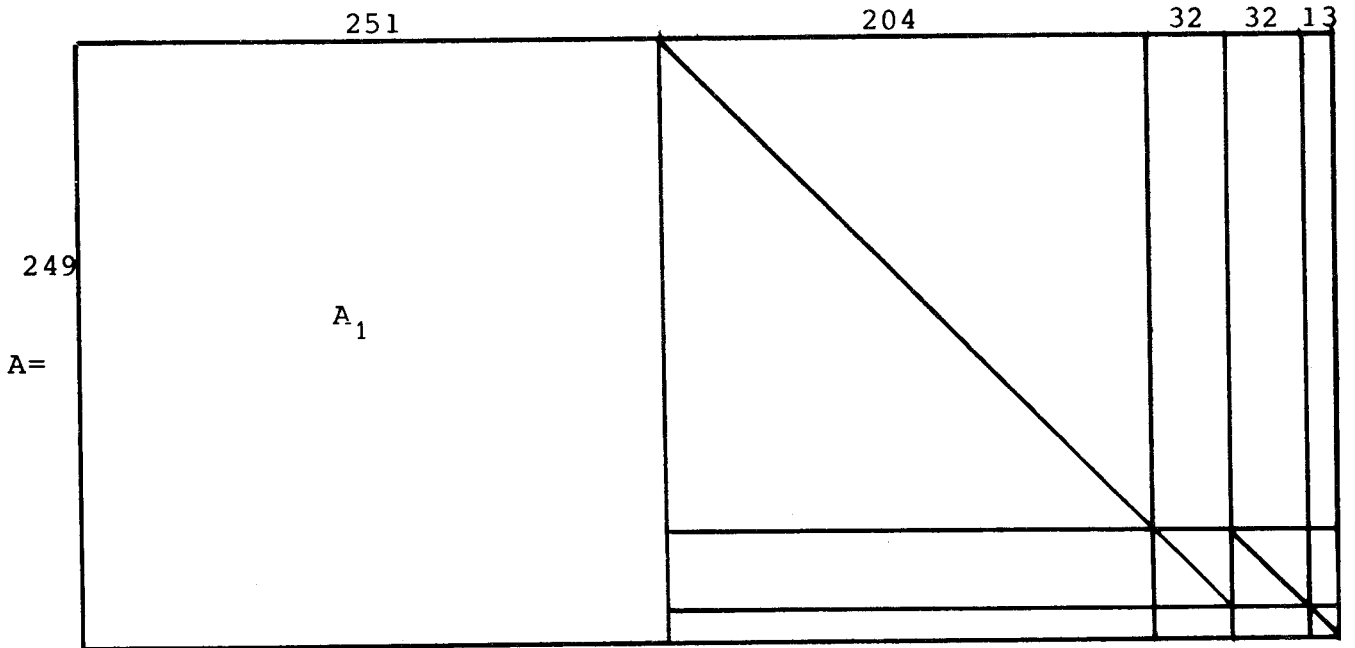
Evolution du nombre de paramètres

Numéro de l'itération	10	20	30	40	50	60	70	80	90	100	112
Nombre de paramètres	0	0	2	3	5	4	5	5	2	2	4

Nombre de paramètres	0	1	2	3	4	5	6
Nombre d'itérations	26	2	9	16	26	18	15

Nombre moyen de paramètres par itération : 3,14

Problème n° 4 : modèle énergétique



$\left\{ \begin{array}{l} 251 \text{ variables} \\ 249 \text{ contraintes} \end{array} \right. \left\{ \begin{array}{l} 204 \text{ en } \leq \\ 32 \text{ en } \geq \\ 13 \text{ en } = \end{array} \right.$   
 $A_1 : 718 \text{ éléments non nuls } (\rho = 1,15\%)$

Problème à traiter :

$\left\{ \begin{array}{l} 532 \text{ variables} \\ 249 \text{ contraintes} \end{array} \right. \left\{ \begin{array}{l} 251 \text{ variables d'origine} \\ 236 \text{ variables d'écart} \\ 45 \text{ variables artificielles} \end{array} \right.$   
 $A : 999 \text{ éléments non nuls } (\rho = 0,75\%)$



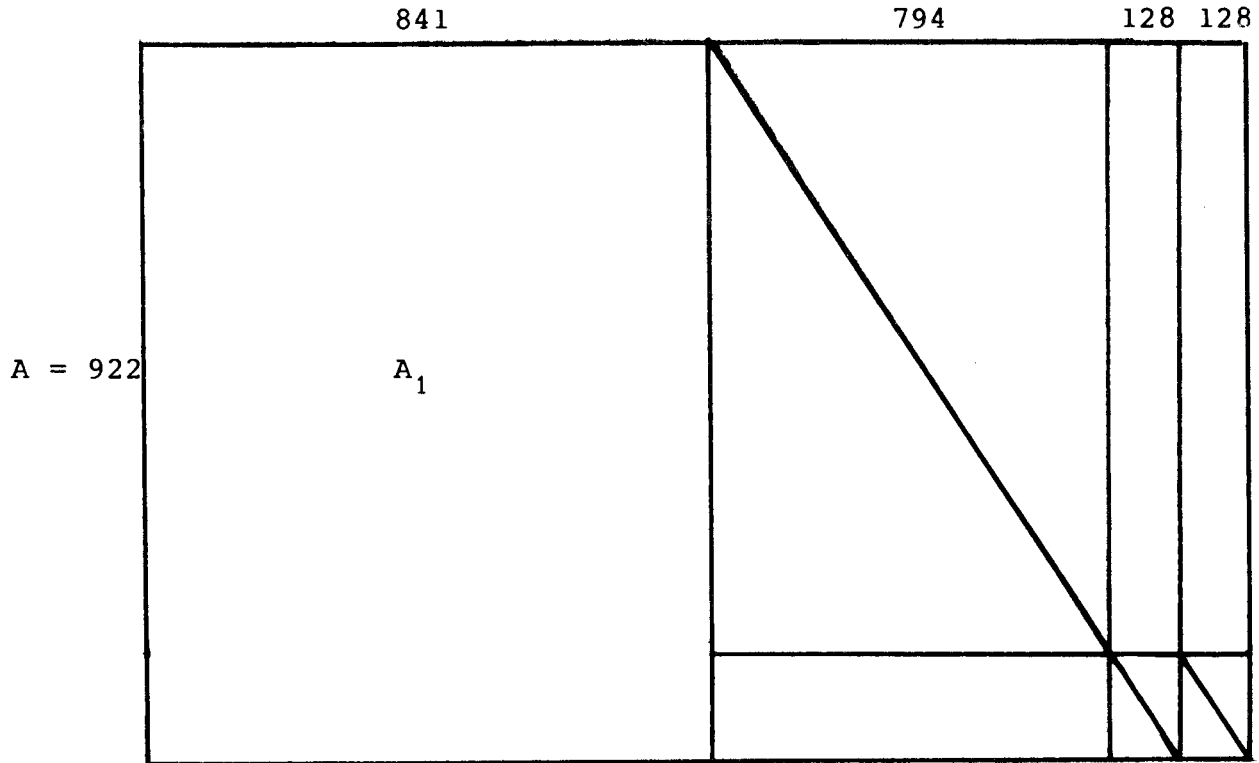
Résultats : (pour la méthode des paramètres)

- 392 itérations
- 1,12 seconde par itération
- 21100 mémoires utilisées (pour un nombre possible de paramètres égal à 10)
- dont 6550 pour les données en double précision
- soit 14650 pour les calculs
  
- le nombre maximum de paramètres est égal à 4.

Nombre de paramètres	0	1	2	3	4
Nombre d'itérations	36	100	112	133	11

Nombre moyen de paramètres par itération : 1,9566

Problème n° 5 : modèle énergétique



$$\left\{ \begin{array}{l} 841 \text{ variables} \\ 922 \text{ contraintes} \end{array} \right\} \begin{cases} 794 \text{ en } \leq \\ 128 \text{ en } \geq \end{cases}$$

A : 2816 éléments non nuls ( $\rho = 0,36\%$ )

Problème à traiter :

$$\left\{ \begin{array}{l} 1891 \text{ variables} \\ 922 \text{ contraintes} \end{array} \right\} \begin{cases} 841 \text{ variables d'origine} \\ 922 \text{ variables d'écart} \\ 128 \text{ variables artificielles} \end{cases}$$

A : 3866 éléments non nuls ( $\rho = 0,22\%$ )



Résultats : (pour la méthode des paramètres)

- 940 itérations
- 11,58 secondes par itération
- 78640 mémoires utilisées (pour un nombre possible de paramètres égal à 10)
- dont 24950 pour les données en double précision
- et 53690 pour les calculs

Le nombre maximum de paramètres est égal à 4.

Nombre de paramètres	0	1	2	3	4
Nombre d'itérations	122	563	220	30	5

Nombre moyen de paramètres par itération : 1,1840

Problèmes n° 6 et 7 :

Les deux programmes linéaires ayant fourni les résultats qui suivent ont été rencontrés lors de la résolution d'un problème de dispatching électrique (réseau grec à 44 noeuds) par la méthode des centres linéarisés (MCL) [3].

On pourra trouver dans [32] une description détaillée des conditions de mise en oeuvre de la MCL et des résultats obtenus

Les matrices de base sont de dimension 89 x 89 ( $\rho \% = 7,5 \%$ )

Programme linéaire 1 :

- 111 itérations
- 0,525 seconde par itération
- le nombre maximum de paramètres rencontré est égal à 9.

Nombre de paramètres	0	1	2	3	4	5	6	7	8	9
Nombre d'itérations	1	5	10	11	10	7	19	17	12	19

Nombre maximum de paramètres par itération : 5,7027

Programme linéaire 2 :

- 157 itérations
- 0,538 seconde par itération
- le nombre maximum de paramètres rencontré est égal à 12.

Nombre de paramètres	0	1	2	3	4	5	6	7	8	9	10	11	12
Nombre d'itérations	1	10	3	8	21	20	9	24	31	17	10	2	1

Nombre moyen de paramètres par itération : 6,2484

C-3- Remarques et commentaires

1 - La résolution des systèmes linéaires par la méthode des paramètres permet de traiter des programmes linéaires à matrice creuse et de grande taille en mémoire centrale.

Le problème n° 5 ne nécessite réellement que 67576 mémoires si on tient compte d'un nombre maximum de paramètres égal à 4.

2 - Le nombre maximum de paramètres reste toujours petit (de l'ordre de 10) pourvu que  $\rho$  le soit.

	Pb 1	Pb 2	Pb 3	Pb 4	Pb 5	Pb 6	Pb 7
$\rho\%$	5,95	3,18	2,79	0,75	0,22	7,5	7,5
Nombre maximum de paramètres	6	8	6	4	4	9	12

Les valeurs 9 et 12 correspondent en fait à de fortes densités (7,5%).

3 - Lorsque la taille du problème augmente, la méthode des paramètres semble être d'un emploi plus avantageux que l'inverse explicite aussi bien en volume de mémoires nécessaires qu'en temps de calcul.

	Rapport des volumes de mémoires *	Rapport des temps moyens par itération
Pb 1	1,89	0,76
Pb 2	2,57	1,08
Pb 3	4,12	1,42

Rapports  $\frac{\text{(inverse explicite)}}{\text{(méthode des paramètres)}}$

\* Ces rapports sont calculés avec un nombre possible de paramètres égal à 10 (en réalité, on a respectivement pour chacun des problèmes 6, 8, 5 paramètres au maximum).

4 - Le temps moyen par itération est surtout fonction de la phase 1 de la méthode ("rangement" de la matrice).

Les problèmes 6 et 7 relatifs au même problème de départ montre que le nombre moyen de paramètres par itération n'a que peu d'influence sur le temps moyen par itération

	Nombre moyen de paramètres	Temps moyen par itération
Pb 6	5,7027	0,525
Pb 7	6,2484	0,538

soit une augmentation moyenne de 0,013 seconde par itération pour une augmentation de 0,5457 du nombre moyen de paramètres (matrice de base de dimension 89 x 89).

Le problème N° 5 est particulièrement significatif à cet effet : le temps moyen de la phase 1 est de 8,51 secondes par itération.

5 - Pour mesurer la précision des résultats obtenus avec la méthode des paramètres, on a calculé la valeur des résidus sur la solution du problème  $\hat{X}$

$$R = |A\hat{X} - B|, \quad R \in \mathbb{R}^m$$

pour les quatre premiers problèmes :  $R_i < 1 \cdot 10^{-12} \quad \forall_i$   
 pour le cinquième :  $R_i < 1 \cdot 10^{-11} \quad \forall_i$

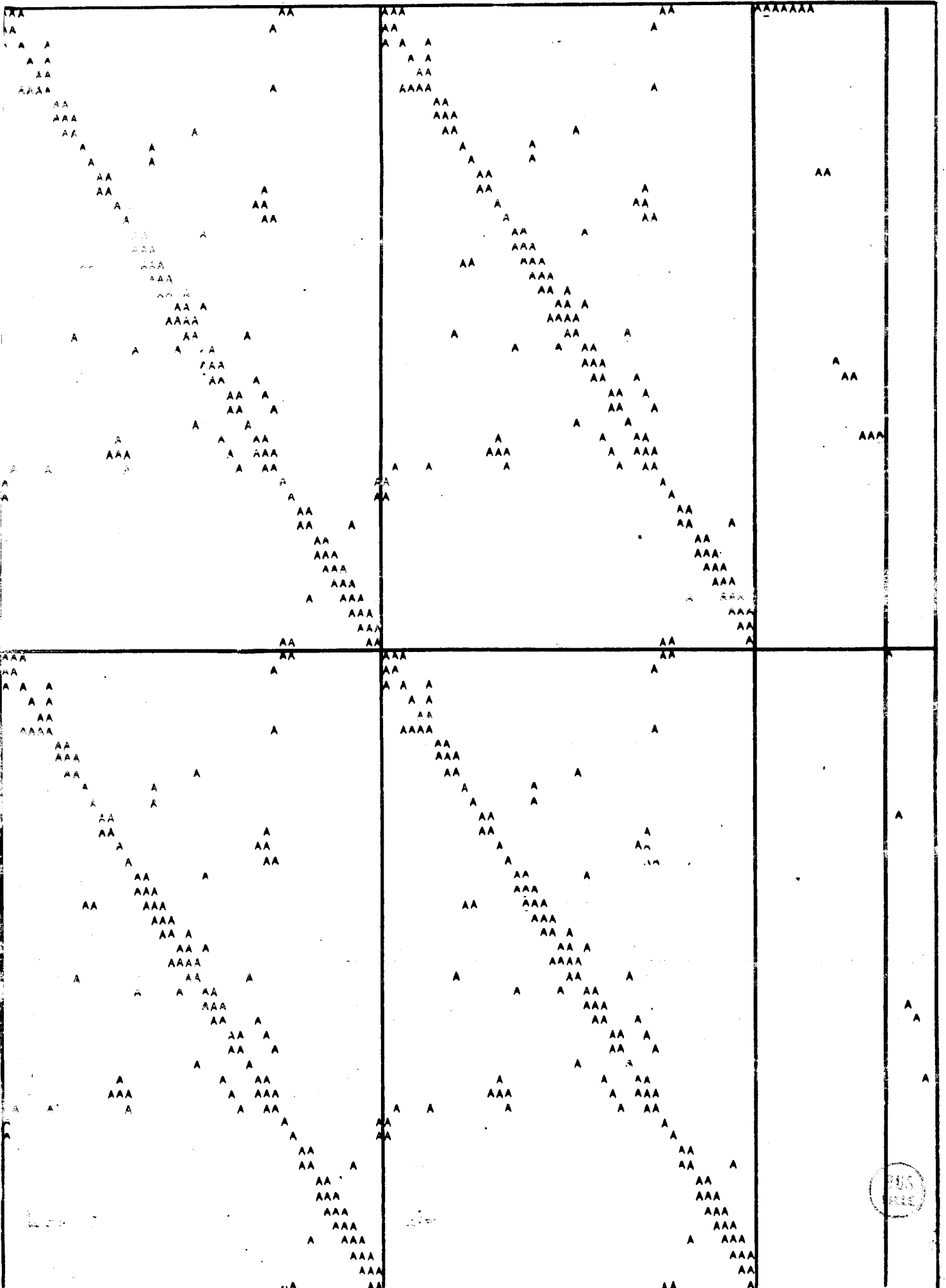
ANNEXE 1 : Matrices de type dispatching

1. Matrice de dispatching du réseau grec
2. Matrice de base la moins creuse
3. Résultat - critère 1
4. Matrice  $A_1$
5. Résultat - critère 4
6. Matrice  $A_2$
7. Résultat - critère 4
8. Matrice  $A_3$
9. Résultat - critère 2
10. Matrice  $A_4$
11. Résultat - critère 2
12. Matrice  $A_5$
13. Résultat - critère 1

$U_i$  (tension)

$-1 - \theta_i$  (phase)

POWER  
 $P_i$  (active)  $Q_i$  (react.)



Grecian Network

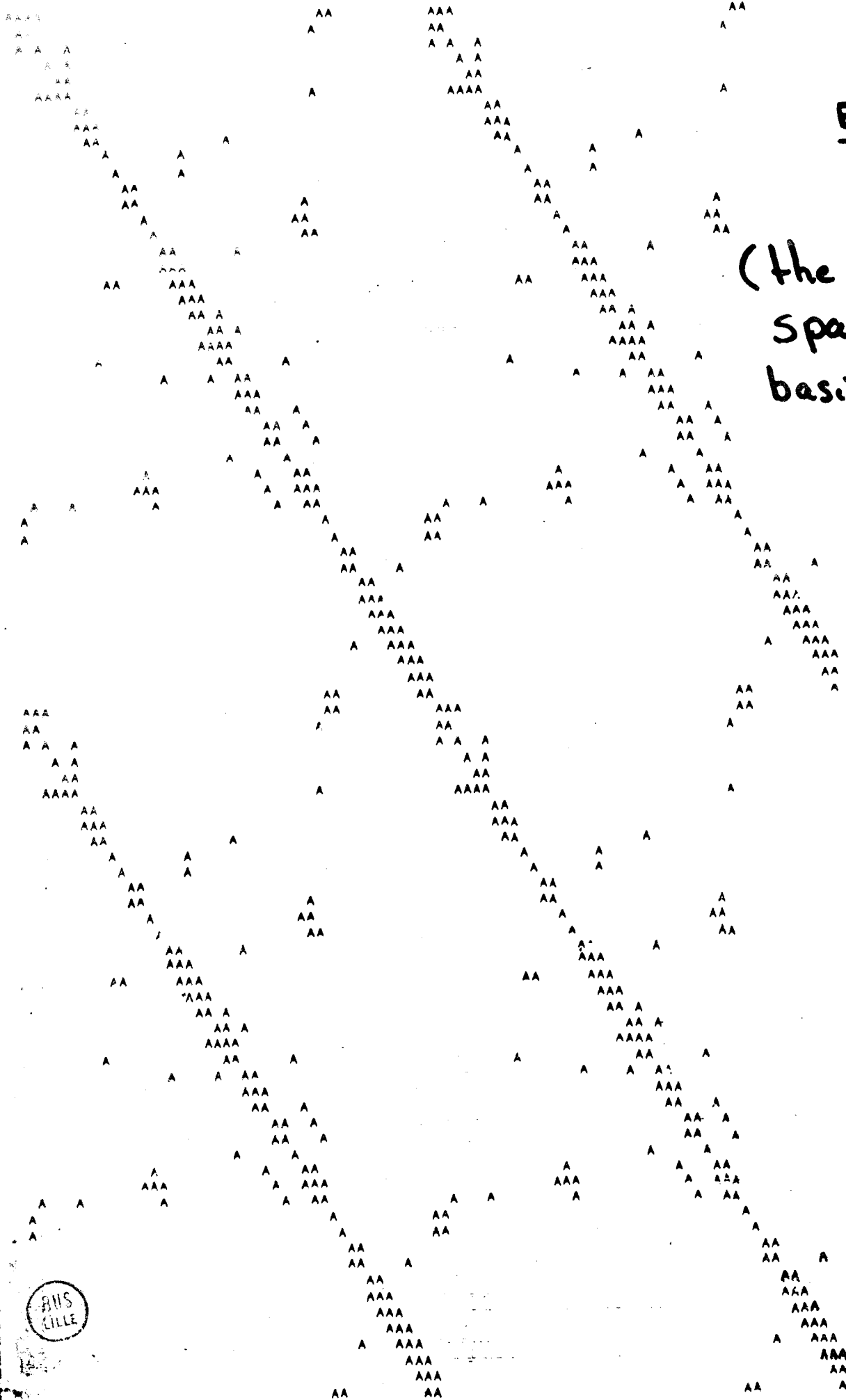


Size  
(88, 88)

ELTS  $\neq 0$

7.24 %

(the least  
sparse  
basic matrix)

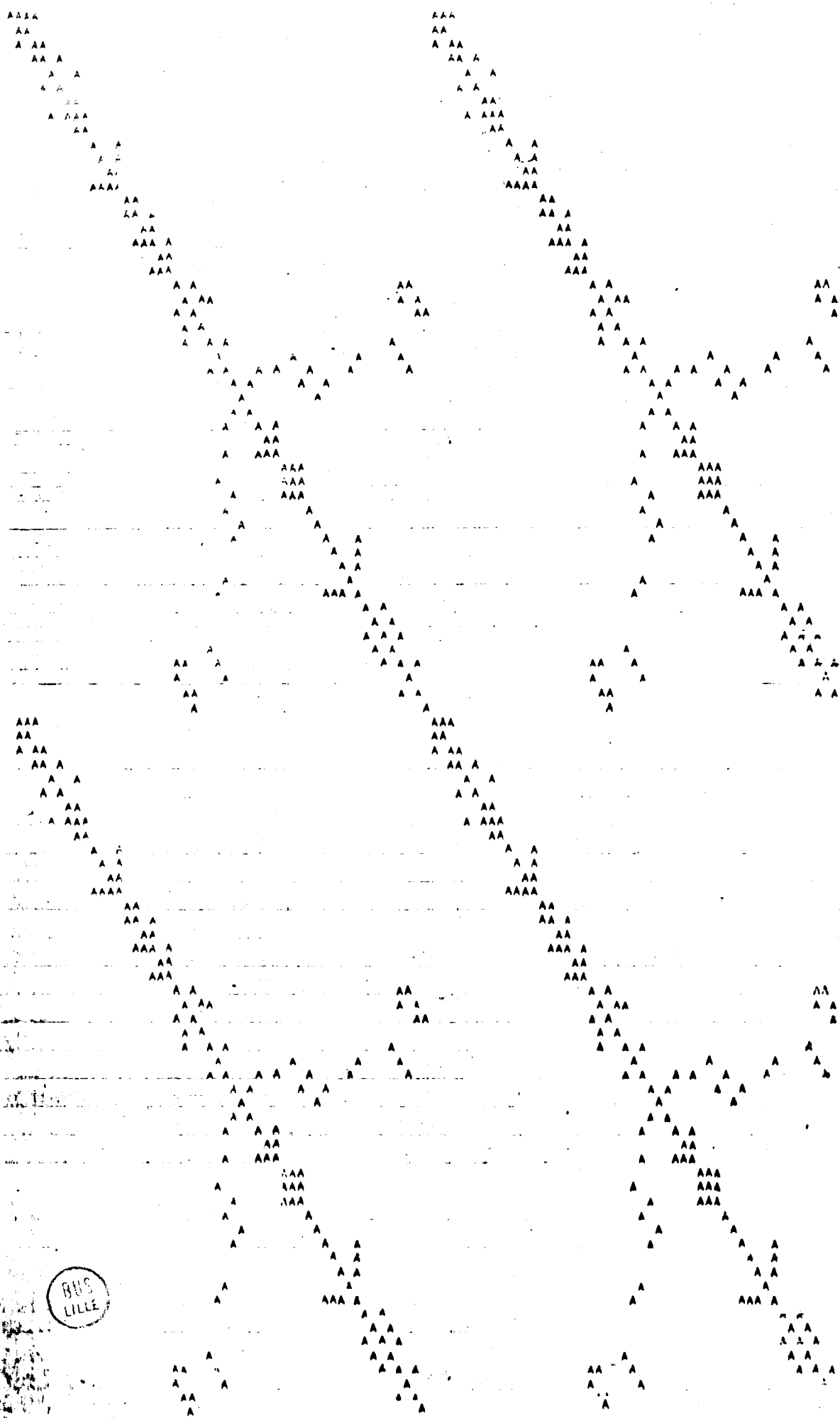


BUS  
LILLE





(100, 10  
5.97 %

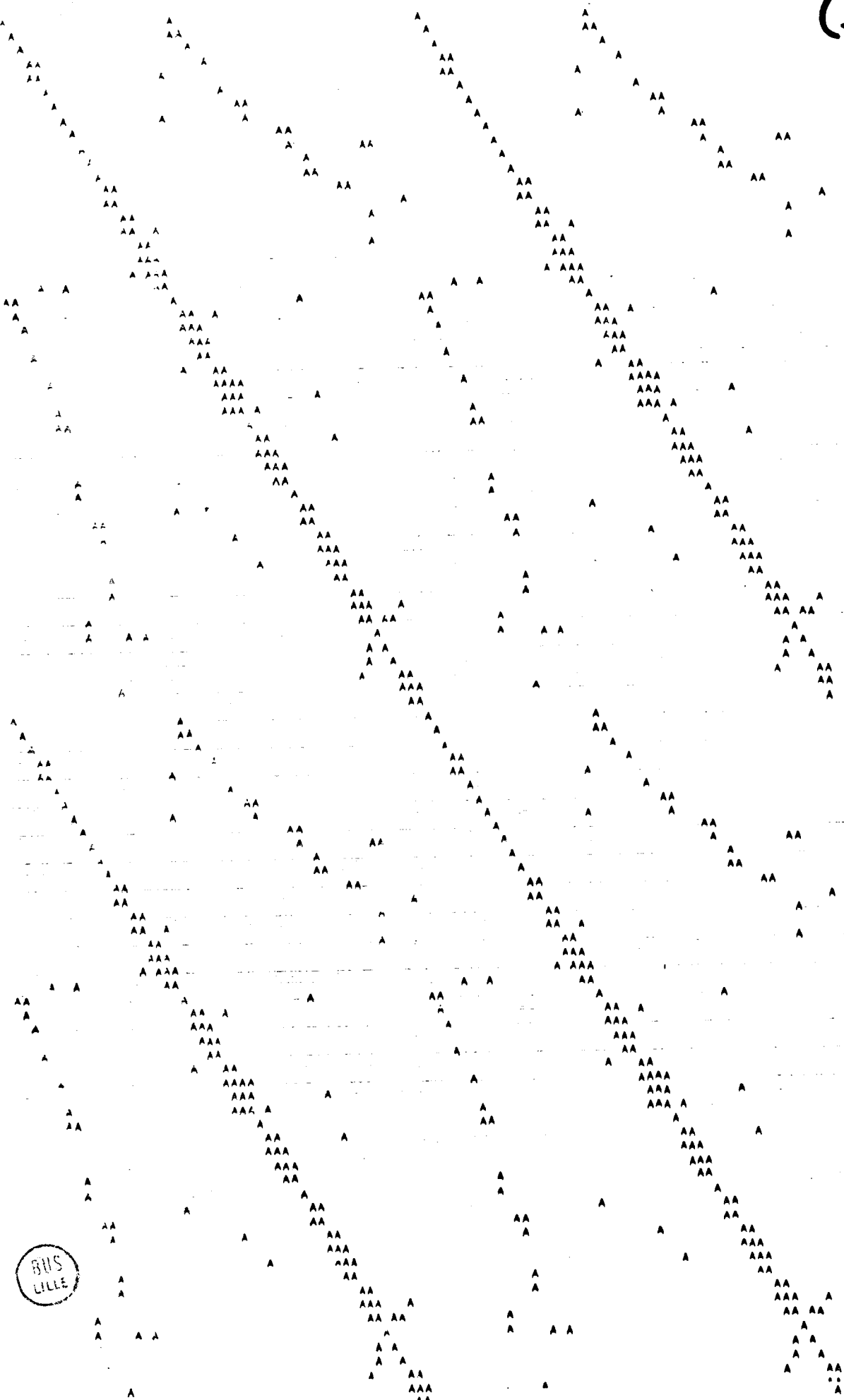


BUS  
LILLE



(100,100

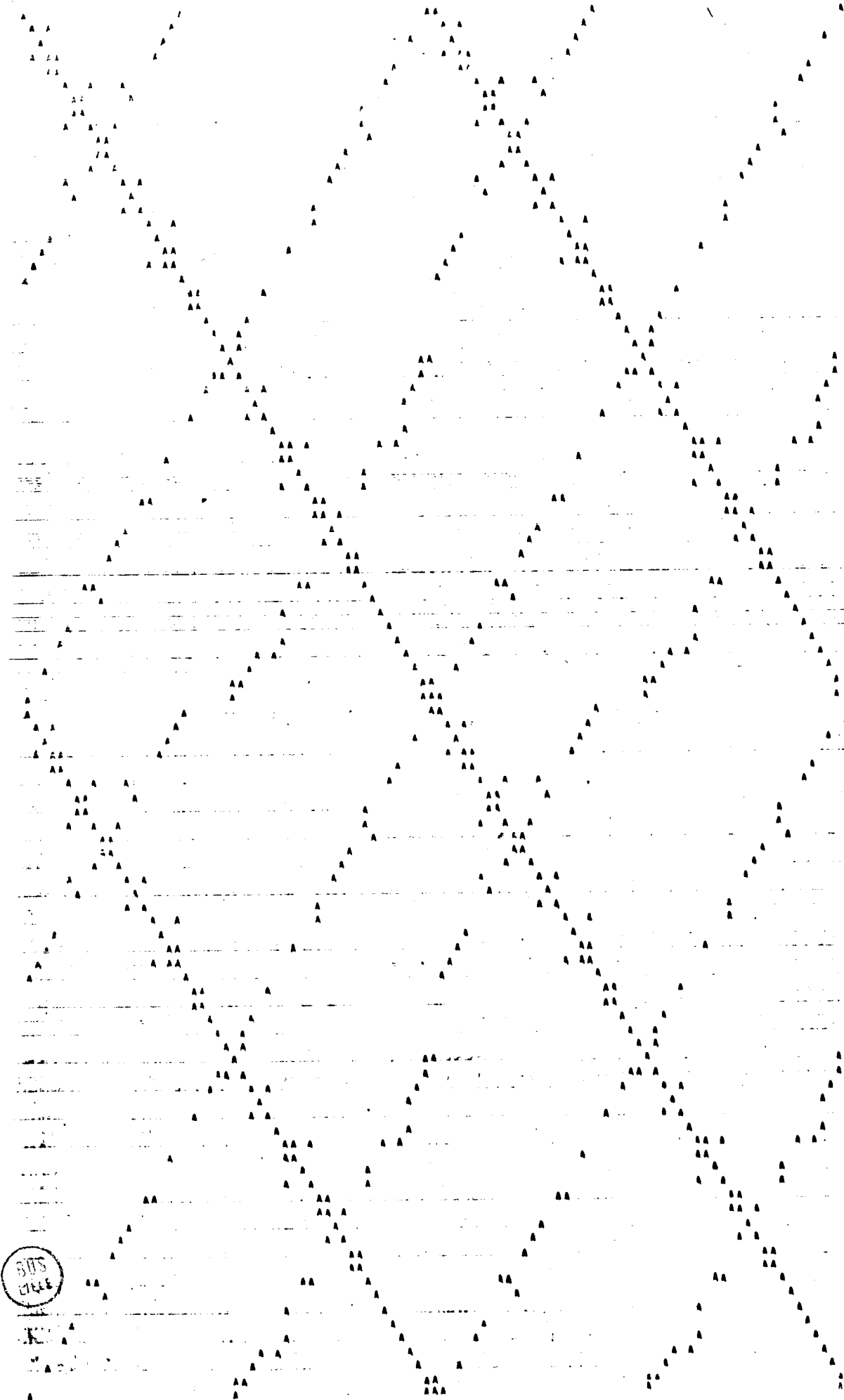
6.49%



BUS  
LILLE



Size  
(100, 10  
ELTS #  
5.93 %



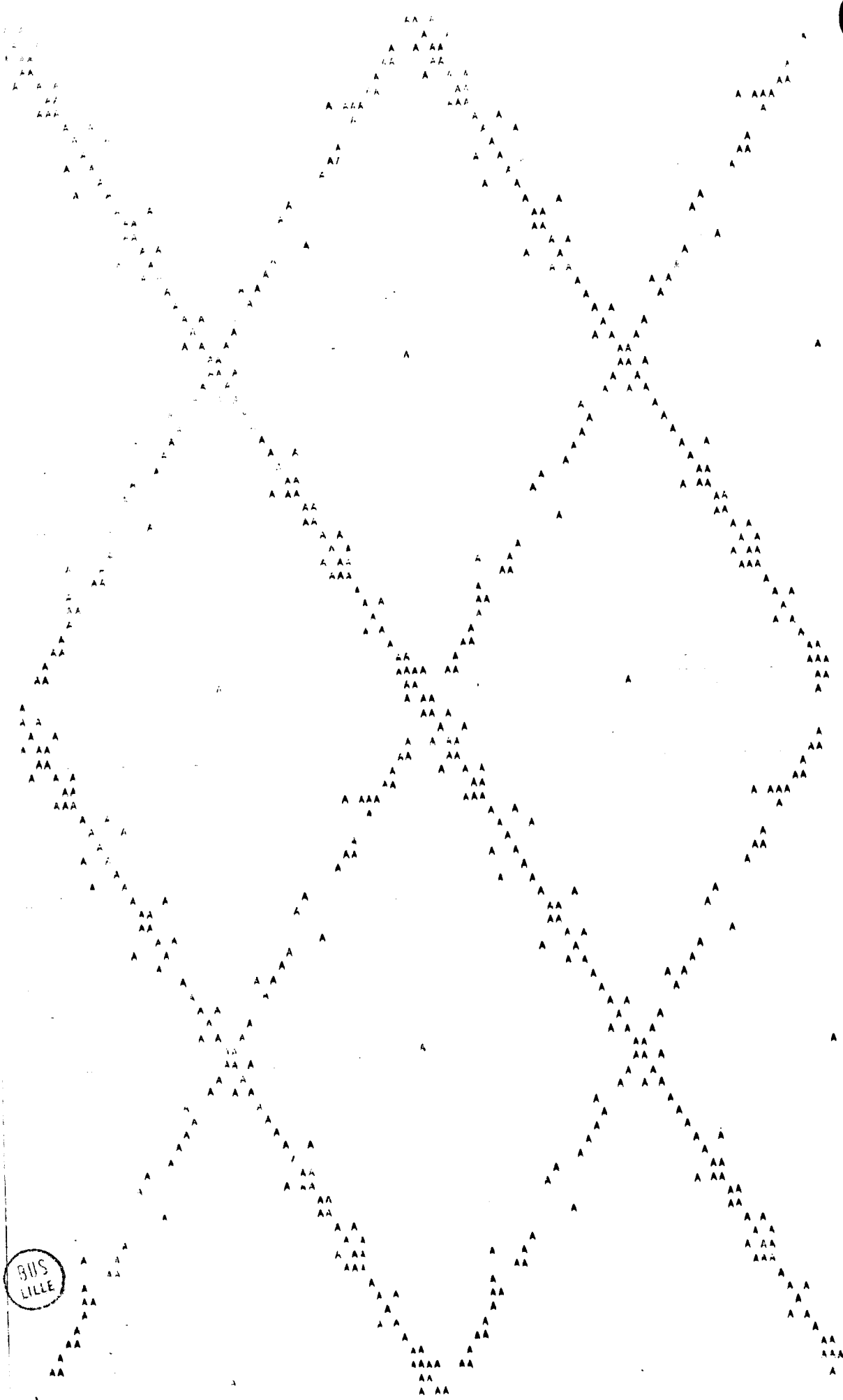
505  
1744

K



(100,100

5.95%



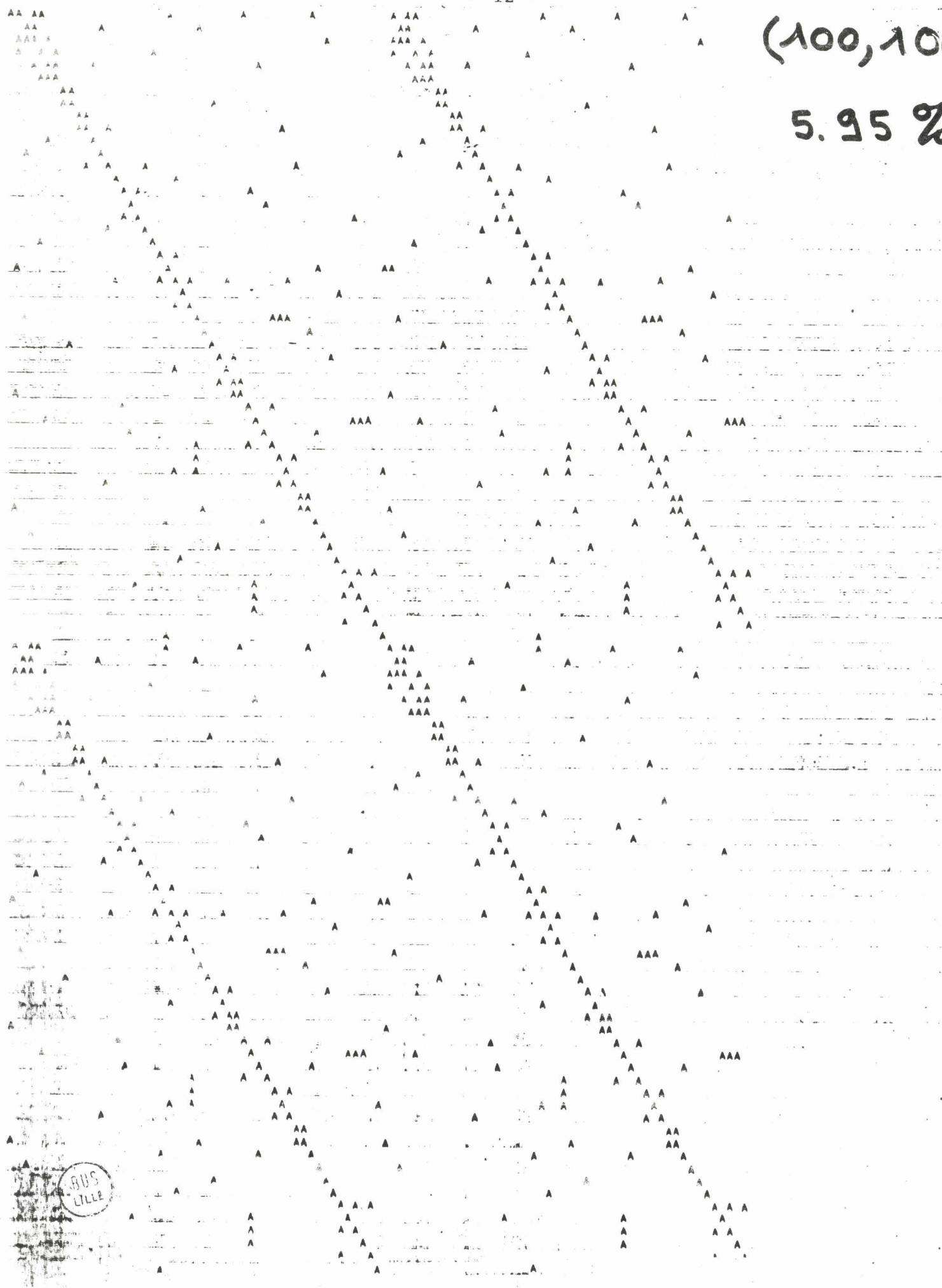
BUS  
LILLE





(100,100)

5.95 %



BUS  
LILLE



## ANNEXE II

### CODE FORTRAN DE PROGRAMMATION DE LA MÉTHODE SIMPLICIALE AVEC RÉOLUTION DES SYSTÈMES LI- NÉAIRES PAR LA MÉTHODE DES PARAMÈTRES.

- \* -

- Les dimensionnements des tableaux sont ceux utilisés lors de la résolution du problème n°4 (chapitre III - page 28).
- Les données sont de la forme :
  - N, M, N1, N2 où N= nombre d'éléments de la matrice A  
M= nombre de contraintes  
N1= nombre de variables d'origine  
N2= nombre de variables d'écart et artificielles
  - IL1 : indices ligne des éléments de A1 considérés colonne par colonne
  - IC1 : indices colonne des éléments de A1 considérés colonne par colonne
  - IC2 : indices colonne des éléments de A1 considérés ligne par ligne
  - A : coefficients des éléments de A1 considérés colonne par colonne

- SM : second membre des contraintes

- FECO : coefficients de la fonction économique.

```

IMPLICIT DOUBLE PRECISION(A-H,O-Z)
INTEGER DC,DCU,DL,BA,HBA
INTEGER RBA
INTEGER DC1,DL1
DIMENSION RBA(532)
DIMENSION DC1(533),DL1(250)
DIMENSION ILR(532),ILU(999),ICU(999),ICR(532),HBA(283)
DIMENSION DC(532),DCU(249),DL(249),IL1(999),IC1(999),IC2(999)
DIMENSION A(999),AU(999),BA(249)
DIMENSION SM(249)
DIMENSION FECO(532),V1(249),V2(249),D(283),PARA(249,10)
  INTEGER P,P1

```

```

READ 9,N,M,N1,N2

```

```

READ 6,(IL1(I),I=1,718)

```

```

READ 6,(IC1(I),I=1,718)

```

```

READ 6,(IC2(I),I=1,718)

```

```

INPUT,(A(I),I=1,718)

```

```

INPUT,(SM(I),I=1,249)

```

```

INPUT,(FECO(I),I=1,251)

```

LECTURE DES DONNEES

```

9  FORMAT(20I5)

```

```

8  FORMAT(4I5)

```

```

6  FORMAT(6I12)

```

```

7  FORMAT(12I5)

```

```

65  FORMAT(8F16.12)

```

```

DO 6050 I=1,249

```

```

6050 DL(I)=0

```

```

  DO 6000 I=1,718

```

```

6000 DL(IL1(I))=DL(IL1(I))+1

```

```

DO 6300 I=1,204

```

```

K=718+I

```

```

A(K)=1

```

```

IL1(K)=I

```

```

IC1(K)=251+I

```

```

6300 CONTINUE

```

```

DO 6100 I=1,32

```

```

K=922+I

```

```

A(K)=-1

```

```

IL1(K)=204+I

```

```

IC1(K)=455+I

```

```

6100 CONTINUE

```

```

DO 6200 I=1,45

```

```

K=954+I

```

```

A(K)=1

```

```

IL1(K)=204+I

```

```

IC1(K)=487+I

```

```

6200 CONTINUE

```

```

J2=0

```

```

J3=0

```

```

DO 6350 I=1,13

```

```

J3=J3+1

```

```

K1=1000-J3

```

```

IC2(K1)=533-I

```

```

DO 6400 J=1,DL(250-I)

```

```

J2=J2+1

```

```

J3=J3+1

```

```

IC2(1000-J3)=IC2(719-J2)

```

```

6400 CONTINUE

```

INTRODUCTION

DES VARIABLES

D'ECART ET ARTIFICIELLES



```

6350 CONTINUE
DO 6500 I=1,32
J3=J3+1
IC2(1000-J3)=520-I
J3=J3+1
IC2(1000-J3)=488-I
DO 6600 J=1,DL(237-I)
J2=J2+1
J3=J3+1
IC2(1000-J3)=IC2(719-J2)

```

```
6600 CONTINUE
```

```

6500 CONTINUE
DO 6700 I=1,204
J3=J3+1
IC2(1000-J3)=456-I
DO 6800 J=1,DL(205-I)
J2=J2+1
J3=J3+1
IC2(1000-J3)=IC2(719-J2)

```

```
6800 CONTINUE
```

```

6700 CONTINUE
OUTPUT J2
OUTPUT J3
DO 6950 I=1,532
ICR(I)=0

```

```

6950 DC(I)=0
DO 6900 I=1,249

```

```

6900 DL(I)=0
DO 5200 I=1,251

```

```

5200 HBA(I)=I
DO 5300 I=1,32

```

```

5300 HBA(251+I)=455+I
DO 5000 I=1,204

```

```

5000 BA(I)=I+251
DO 5100 I=1,45

```

```

5100 BA(204+I)=I+487
DO 5500 I=1,45

```

```

5500 FECO(487+I)=-500000
DO 5620 I=1,236

```

```

5620 FECO(251+I)=0
OUTPUT N
OUTPUT M
OUTPUT N1
OUTPUT N2
CALL UCTIME(0,INTER)

```

```

DO 110 J=1,N
DC(IC1(J))=DC(IC1(J))+1
110 DL(IL1(J))=DL(IL1(J))+1
DC1(N1+N2+1)=N+1
DL1(M+1)=N+1
DO 120 I=1,(N1+N2)
J=N1+N2+1-I
RBA(J)=0

```

```

120 DC1(J)=DC1(J+1)-DC(J)
DO 130 I=1,M
J=M+1-I

```

```

130 DL1(J)=DL1(J+1)-DL(J)
L8=0

```

```

1 CONTINUE
L8=L8+1

```

```

DO 145 J=1,M
145 DL(J)=DL1(J+1)-DL1(J)
DO 11 J=1,M

```

```

11 DCU(J)=DC(BA(J))
DO 147 J=1,(N1+N2)

```

```

147 RBA(J)=0

```

DETERMINATION DE  
LA BASE ET DE L'HORS-BASE  
DE DEPART

CALCUL DES DEGRES  
ET DEGRES CUMULES  
DES LIGNES ET DES  
COLONNES DE A

```
DO 146 J=1,M
146 RBA(BA(J))=J
J2=0
J4=1
DO 100 J1=1,M
K=M+2
K2=0
DO 12 J=1,M
IF(K-DCU(J))12,14,15
15 K=DCU(J)
14 K1=J
12 CONTINUE
ILR(J1)=BA(K1)
K2=DC1(BA(K1))-1
17 CONTINUE
K5=DC(BA(K1))
DO 19 J=1,K5
AU(J2+J)=A(K2+J)
ILU(J2+J)=IC1(K2+J)
ICU(J2+J)=IL1(K2+J)
K6=ICU(J2+J)
IF(ICR(K6).NE.0)GOTO 19
ICR(K6)=J4
J4=J4+1
K3=DL1(K6)-1
21 CONTINUE
DO 23 J3=1,DL(K6)
L7=RBA(IC2(K3+J3))
IF(L7.EQ.0) GOTO 23
DCU(L7)=DCU(L7)-1
23 CONTINUE
19 CONTINUE
DCU(K1)=M+1
J2=J2+DC(BA(K1))
100 CONTINUE
DO 150 J=1,M
BA(J)=ILR(J)
DCU(J)=DC(BA(J))
ICR(J)=0
150 V1(J)=FECO(BA(J))
1000 CONTINUE
```

RECHERCHE DE LA  
PERMUTATION ET  
REEMPLISSAGE DES  
TABLEAUX AUXILIAIRES

---

```
J5=0
P=0
P1=0
J8=0
DO 250 J=1,M
IF(P1-P)270,271,271
270 P1=P
271 CONTINUE
K7=0
DO 203 J1=1,DCU(J)
IF(ICR(ICU(J5+J1)).NE.0)GOTO 203
K7=K7+1
ILR(K7+J8)=ICU(J5+J1)
203 CONTINUE
J8=J8+K7
DO 204 J1=1,DCU(J)
J7=ICR(ICU(J5+J1))
IF(J7.EQ.0)GOTO 204
IF(J7.GT.M)GOTO 206
IF(P.EQ.0)GOTO 260
DO 207 J6=1,P
207 PARA(J,J6)=PARA(J,J6)-PARA(J7,J6)*AU(J5+J1)
260 V1(J)=V1(J)-V1(J7)*AU(J5+J1)
V2(J)=V2(J)-V2(J7)*AU(J5+J1)
GOTO 204
```



```

206 J7=J7-M
    PARA(J,J7)=PARA(J,J7)-AU(J5+J1)
204 CONTINUE
    IF(K7.EQ.0) GOTO 210
    DO 257 J1=1,DCU(J)
    J7=ICR(J5+J1)
    IF(ICR(J7).NE.0) GOTO 257
    IF(K7.EQ.1)GOTO 258
    P=P+1
    K7=K7-1
    PARA(J,P)=-AU(J5+J1)
    ICR(J7)=M+P
257 CONTINUE
258 CONTINUE
    IF(P.EQ.0) GOTO 255
    DO 259 J6=1,P
259 PARA(J,J6)=PARA(J,J6)/AU(J5+J1)
255 V1(J)=V1(J)/AU(J5+J1)
    V2(J)=V2(J)/AU(J5+J1)
    ICR(J7)=J
    J5=J5+DCU(J)
    GOTO 250
210 CONTINUE
    IF(P.EQ.1) GOTO 288
280 CONTINUE
    T2=0
    DO 281 J6=1,P
    IF(ABS(PARA(J,J6))-T2)281,281,282
    T2=ABS(PARA(J,J6))
    J1=J6
281 CONTINUE
    DO 283 J6=1,(N1+N2)
    IF(ICR(J6).EQ.(M+J1))GOTO 284
283 CONTINUE
284 I=J6
    DO 285 J6=1,(N1+N2)
    IF(ICR(J6).EQ.(M+P)) GOTO 286
285 CONTINUE
286 ICR(J6)=M+J1
    ICR(I)=M+P
    DO 287 J6=1,J
    T2=PARA(J6,P)
    PARA(J6,P)=PARA(J6,J1)
    PARA(J6,J1)=T2
287 CONTINUE
288 CONTINUE
    IF(L.EQ.1)GOTO 1050
    DO 211 J1=1,M
    IF(ICR(J1).EQ.(M+P))GOTO 212
211 CONTINUE
212 ICR(J1)=J
    GOTO 1060
1050 CONTINUE
    DO 1051 J1=1,(N1+N2)
    IF(ICR(J1).EQ.(M+P))GOTO 1052
1051 CONTINUE
1052 ICR(J1)=J
1060 CONTINUE
    IF(P.EQ.1)GOTO 261
    DO 213 J1=1,(P-1)
213 PARA(J,J1)=-PARA(J,J1)/PARA(J,P)
261 V1(J)=-V1(J)/PARA(J,P)
    V2(J)=-V2(J)/PARA(J,P)
    DO 214 J1=1,(J-1)
    IF(PARA(J1,P).EQ.0) GOTO 214
    IF (P.EQ.1) GOTO 216

```

RESOLUTION DES  
SYSTEMES LINEAIRES  
PAR LA METHODE  
DES PARAMETRES



```
DO 215 J6=1,(P-1)
215 PARA(J1,J6)=PARA(J1,J6)+PARA(J1,P)*PARA(J,J6)
V1(J1)=V1(J1)+PARA(J1,P)*V1(J)
V2(J1)=V2(J1)+PARA(J1,P)*V2(J)
PARA(J1,P)=0
GOTO 214
216 CONTINUE
V1(J1)=V1(J1)+PARA(J1,1)*V1(J)
V2(J1)=V2(J1)+PARA(J1,1)*V2(J)
PARA(J1,1)=0
214 CONTINUE
PARA(J,P)=0
P=P-1
J5=J5+DCU(J)
250 CONTINUE
```

---

```
OUTPUT P1
IF(L.EQ,1)GOTO 1100
DO 299 J1=1,(N1+N2-M)
299 D(J1)=0
DO 300 J1=1,(N1+N2-M)
K1=DC1(HBA(J1))-1
301 CONTINUE
DO 303 J3=1,DC(HBA(J1))
303 D(J1)=D(J1)+A(K1+J3)*V1(ICR(IL1(K1+J3)))
D(J1)=FECO(H3A(J1))-D(J1)
300 CONTINUE
J9=1
```

CALCUL DU  
CRITERE DE  
CANDIDATURE

---

```
T2=0,000000001
DO 320 J1=1,(N1+N2-M)
IF (L9.EQ,0) GOTO 323
IF(HBA(J1).GE.210) GOTOJ20
323 CONTINUE
IF(T2-D(J1))322,320,320
322 J9=0
IS=HBA(J1)
T2=D(J1)
320 CONTINUE
DO 401 J=1,M
```

RECHERCHE DE  
LA VARIABLE ENTRANT  
DANS LA BASE

---

```
401 DL(J)=0
DO 400 J=1,J2
400 DL(ICU(J))=DL(ICU(J))+1
K=1
DO 402 J=1,M
J1=M+1-J
ICR(ILR(J1))=K
K=K+DL(ILR(J1))
402 CONTINUE
DO 410 J=1,M
J1=M+1-J
K=DC1(BA(J1))-1
411 CONTINUE
DO 413 J3=1,DC(BA(J1))
J4=K+J3
J5=ICR(IL1(J4))
ICU(J5)=IC1(J4)
AU(J5)=A(J4)
ILU(J5)=IL1(J4)
413 ICR(IL1(J4))=ICR(IL1(J4))+1
410 CONTINUE
```

TRANSPOSITION DE  
LA MATRICE DE  
BASE

---

```
DO 500 J=1,M
V1(J)=0
500 V2(J)=0
IF(J9.EQ,1)GOTO 510
K=DC1(IS)-1
501 CONTINUE
DO 503 J=1,DC(IS)
```



```

503 V2(IL1(K+J))=A(K+J)
DO 504 J=1,M
504 V1(J)=V2(ILR(M+1-J))
510 CONTINUE
DO 505 J=1,M
505 V2(J)=SM(ILR(M+1-J))
DO 506 J=1,(N1+N2)
506 ICR(J)=0
DO 507 J=1,M
507 DCU(J)=DL(ILR(M+1-J))
L=1
GOTO 1000

```

DETERMINATION DES  
SECONDS MEMBRES DES  
DEUX SYSTEMES A RESOUDRE  
SIMULTANEMENT

```

1100 CONTINUE
L=0
IF(J9.EQ.1)GOTO 2102
T2=100000000
J1=0
E=0.000001
DO 600 J=1,M
IF(V1(J).LE.E)GOTO 600
IF(V2(J).LT.(-E)) GOTO 600
T1=V2(J)/V1(J)
IF(T2-T1) 600,601,601
601 CONTINUE
T2=T1
J1=J
600 CONTINUE

```

RECHERCHE DE LA  
VARIABLE SORTANT  
DE LA BASE

```

IF(J1.EQ.0) GOTO 2000
DO 610 J=1,(N1+N2)
IF(ICR(J).EQ.J1)GOTO 611
610 CONTINUE
611 IR=J
DO 700 J=1,M
IF(BA(J).EQ.IR)GOTO 701
700 CONTINUE
701 BA(J)=IS
DO 702 J=1,(N1+N2-M)
IF(HBA(J).EQ.IS)GOTO 703
702 CONTINUE
703 HBA(J)=IR

```

MISE A JOUR DES  
TABLEAUX BASE ET  
HORS - BASE

```

2102 CONTINUE
SO=0
DO 2101 J=1,(N1+N2)
IF(ICR(J).EQ.0) GOTO 2101
SO=SO+V2(ICR(J))*FECO(J)
2101 CONTINUE
OUTPUT SO
OUTPUT IS
OUTPUT IR
IF(J9.EQ.1)GOTO 2100
DO 707 J=1,(N1+N2)
707 ICR(J)=0
GOTO 1

```

CALCUL DE LA  
SOLUTION : VALEUR  
DE LA FONCTION ECONOMI  
QUE

```

2000 SO=1000
OUTPUT SO
2100 CONTINUE
CALL UCTIME(1,INTER)
OUTPUT INTER
PRINT9,(BA(I),I=1,M)
PRINT9,(HBA(I),I=1,(N1+N2-M))
OUTPUT SO
PRINT 9,(ICR(I),I=1,(N1+N2))
DO2450 I=1,(N1+N2)
IF(ICR(I).EQ.0) GOTO 2455
V=V2(ICR(I))
GOTO 2460

```

IMPRESSIONS  
DES RESULTATS

```
2455 V=0
2460 OUTPUT I;
      OUTPUT V
2450 CONTINUE
      PRINT 65,(FECO(I),I=1,(N1+N2))
      PRINT 65,(V1(I),I=1,M)
      PRINT 66,(D(I),I=1,283)
  66  FORMAT(8F16.8)
2500 END
```



## BIBLIOGRAPHIE.

1. GASTINEL  
Analyse numérique linéaire - Hermann
2. WILKINSON  
The algebraic eigenvalue problem - Oxford University Press, 1965.
3. NOTE D'ÉTUDE AN-10  
Résolution des systèmes linéaires - Centre de calcul scientifique de l'armée - Arcueil- Juillet 1970.
4. CLAUDE BREZINSKI  
Résolution des systèmes d'équations linéaires - Laboratoire de Calcul de Lille I.
5. ADO W. POOCH ET AL. NIEDER  
A survey of indexing techniques for sparse matrices  
Computing survey, vol 5, n° 2, Juin 1973, p. 110.
6. CARPENTIER J.  
Elimination ordonnée - Note HR 10330/3
7. SCOTT B. ET HOBSON E.  
Solution of large power system Networks by ordered  
Elimination - IEE Proc., Vol 118, 1971, pp 125-134.
8. CARPENTIER J.  
Un processus diminuant le volume des calculs dans la  
résolution des systèmes linéaires à matrices creuses.  
3e congrès de calcul et de traitement de l'information  
A.F.C.A.L.T.I., Dunod 1965, pp. 63-71.

9. MARKOWITZ H.M.  
The elimination form of the inverse and its application  
to linear programming  
Management Sciences 3 - 1957 - pp. 255-269.
10. DUFF L.S. ET REID J.K.  
A comparison of sparsity orderings for obtaining a  
pivotal sequence in gaussian elimination.  
Journal of the Institute of Mathematics and its Applica-  
tions 14 - 1974 - pp 281-291.
11. TEWARSON R.P.  
Solution of a system of simultaneous linear equations  
with a sparse coefficient matrice by elimination methods  
BIT, 7 (1967) pp 226-239.
12. GUSTAVSON G.  
Some basic techniques for solving sparse systems of linear  
equations, "Sparse matrices and their applications",  
PLENUM 1972.
13. SATO N. ET TINNEY W.F.  
Techniques for exploiting the sparsity of the network  
admittance matrix.  
IEEE trans. PAS (1963), pp 944-950.
14. HARARY F.  
Sparse matrices and graph theory. Large sparse sets of  
linear equations.  
REID - 1971 - Academic Press.
15. TEWARSON R.P.  
Row-Column permutation of sparse matrices,  
Comput. J. 10 (1967) pp 128-135.

16. TEWARSON R.P.  
Sorting and ordering sparse linear systems in "large sparse sets of linear equations".  
REID - 1971 - Academic Press.
17. ALWAY G.G. ET MARTIN D.W.  
An algorithm for reducing the band width of a matrix of symmetrical configuration.  
Comput. J. 8 (1965) pp 264-272.
18. CUTHILL E. ET MC KEE J.  
Reducing the band width of sparse symmetric matrices.  
Applied Mathematics Laboratory, Naval Slip.  
Research and Development Center, Washington, D.C., 1969.
19. STEWARD DONALD V.  
Partitioning and Teoring Systems of equations.  
SIAM Journal on Numerical Analysis 2 (1965) pp 345-365.
20. LIN T.D. ET MAH R.S.H.  
Hierarchical partition  
Mathematical Programming 12 (1977) pp 261-278.
21. TEWARSON R.P.  
Computations with sparse matrices.  
SIAM Review, Vol 13, n° 4, octobre 1970.
22. ZOLLENKOPF K.  
Bi Factorisation - Basic computational Algorithm and Programming Techniques.  
In "Large sparse set of linear equations" REID - 1971 - Academic Press.
23. HUARD PIERRE  
Problèmes de mémorisation soulevés par la résolution numérique des grands problèmes d'optimisation.  
Compte-rendu du Colloque d'Analyse numérique de Port-Bail, 1976.

24. MAURRAS J.F.  
Optimisation of the flow through Networks with gains.  
Mathematical Programming - vol 3, N° 2, Oct. 1972,  
pp 135-144.
25. RYSER M.J.  
Combinatorial Mathematics.  
The Mathematical Association of America - 1963.
26. DANTZIG G.B.  
Applications et prolongements de la programmation linéaire.  
DUNOD, 1966.
27. TOLLA P.  
Amélioration de la rapidité d'exécution de méthodes simplificiales en programmation mathématique.  
Avril 1977.
28. HUARD PIERRE  
Analyse numérique linéaire - Optimisation en contraintes linéaires  
Laboratoire de Calcul UER IEEA - LILLE I.
29. HEURGON E.  
Programmation linéaire en nombres entiers, méthode arborescente de R. Faure et Y. Malgranche.  
Thèse de 3e cycle - 1967 - PARIS VII.
30. CHARNES A. ET RAIKE W.M, STUTZ J.D, WALTERS A.S.  
On generation of test problems for linear programming codes.  
Communications of the ACM, vol 17, n° 10, octobre 1974.

31. HUARD PIERRE

Programmation mathématique convexe.

Bulletin de la Direction des Etudes et Recherches EDF

Série C, 1968, pp 61-74.

32. DENEL JACQUES

Communication au Xe Symposium on "Mathematical Programming",

Montréal - 1979.

