



HAL
open science

Étude de quelques méthodes de calcul du polynôme caractéristique et des valeurs propres

Jean-Louis Guyot

► **To cite this version:**

Jean-Louis Guyot. Étude de quelques méthodes de calcul du polynôme caractéristique et des valeurs propres. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1969. Français. NNT : . tel-00281660

HAL Id: tel-00281660

<https://theses.hal.science/tel-00281660>

Submitted on 23 May 2008

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

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

N° d'ordre

T H E S E

présentée à
LA FACULTE DES SCIENCES DE L'UNIVERSITE DE GRENOBLE

pour obtenir

LE GRADE DE DOCTEUR DE TROISIEME CYCLE
"MATHEMATIQUES APPLIQUEES"

par

Jean Louis GUYOT

Licencié-es-Sciences

ETUDE DE QUELQUES METHODES DE CALCUL DU POLYNOME

CARACTERISTIQUE ET DES VALEURS PROPRES

Thèse soutenue le 15 mars 1969, devant la Commission d'Examen :

MM. J. KUNTZMANN	Président
N. GASTINEL	Examineur
P.J. LAURENT	Examineur

N° d'ordre

T H E S E

présentée à
LA FACULTE DES SCIENCES DE L'UNIVERSITE DE GRENOBLE

pour obtenir

LE GRADE DE DOCTEUR DE TROISIEME CYCLE
"MATHEMATIQUES APPLIQUEES"

par

Jean Louis GUYOT

Licencié-es-Sciences

ETUDE DE QUELQUES METHODES DE CALCUL DU POLYNOME

CARACTERISTIQUE ET DES VALEURS PROPRES

Thèse soutenue le 15 mars 1969, devant la Commission d'Examen

MM. J. KUNTZMANN	Président
N. GASTINEL	Examineur
P.J. LAURENT	Examineur

PROFESSEURS TITULAIRES(suite)

MM.	KLEIN J.	Mathématiques
	VAILLANT F.	Zoologie et Hydrobiologie
	ARNAUD Paul	Chaire de Chimie
	SENGEL P.	Chaire de Zoologie
	BARNOUD F.	Chaire de Biozyntèse de la Cellulose
	BRISSONNEAU P.	Physique
	GAGNAIRE	Chaire de Chimie Physique
Mme	KOFLER L.	Botanique
	DEGRANGE Charles	Zoologie
	PEBAY-PEROULA J.C.	Physique
	RASSAT A.	Chaire de Chimie Systématique
	DUCROS P.	Chaire de Cristallographie Physique
	DODU Jacques	Chaire de Mécanique Appliquée I.U.T.
	ANGLES D'AURIAC P.	Mécanique des Fluides
	LACAZE A.	Thermodynamique

PROFESSEURS SANS CHAIRE

MM.	GIDON P.	Géologie et Minéralogie
	GIRAUD P.	Géologie
	PERRET R.	Servomécanisme
Mme	BARBIER M.J.	Electrochimie
Mme	SOUTIF J.	Physique
	COHEN J.	Electrotechnique
	DEPASSEL R.	Mécanique des Fluides
	GASTINEL N.	Mathématiques Appliquées
	GLENAT R.	Chimie
	BARRA J.R.	Mathématiques Appliquées
	COUMES A.	Electronique
	PERRIAUX J.	Géologie et Minéralogie
	ROBERT A.	Chimie Papetière
	BIAREZ J.P.	Mécanique Physique
	BONNET G.	Electronique
	CAUQUIS G.	Chimie Générale
	BONNETAIN L.	Chimie Minérale
	DEPOMMIER P.	Etude Nucléaire et Génie Atomique
	HACQUES Gérard	Calcul Numérique
	POLOUJADOFF M.	Electrotechnique

PROFESSEURS ASSOCIES

MM.	NAPP-ZINN	Botanique
	RODRIGUES Alexandre	Mathématiques Pures
	STANDING Kenneth	Physique Nucléaire

MAITRES DE CONFERENCES

MM.	LANCIA Roland	Physique Atomique
Mme	KAHANE J.	Physique
	DEPORTES C.	Chimie
Mme	BOUCHE L.	Mathématiques
	SARROT-REYNAUD	Géologie Propédeutique

MAITRES DE CONFERENCES (suite)

Mme	BONNIER M.J.	Chimie
MM.	KAHANE A.	Physique Générale
	DOLIQUE J.M.	Electronique
	BRIERE G.	Physique M.P.C.
	DESRE G.	Chimie S.P.C.N.
	LAJZROWICZ J.	Physique M.P.C.
	VALENTIN P.	Physique M.P.C.
	BERTRANDIAS J.P.	Mathématiques Appliquées T.M.P.
	LAURENT P.J.	Mathématiques Appliquées T.M.P.
	CAUBET J.P.	Mathématiques Pures
	PAYAN J.J.	Mathématiques
Mme	BERTRANDIAS F.	Mathématiques Pures M.P.C.
	LONGEGUEUE J.P.	Physique
	NIVAT M.	Mathématiques Appliquées
	SOHM J.C.	Electrochimie
	ZADWORNY F.	Electronique
	DURAND F.	Chimie Physique
	CARLER G.	Biologie Végétale
	AUBERT G.	Physique M.P.C.
	DELPUECH J.J.	Chimie Organique
	PFISTER J.C.	Physique C.P.E.M.
	CHIBON P.	Biologie Animale
	IDELMAN S.	Physiologie Animale
	BOUVARD Maurice	Hydrologie
	RICHARD Lucien	Botanique
	PELMONT Jean	Physiologie Animale
	BLOCH D.	Electrotechnique I.P.
	BOUSSARD J.Claude	Mathématiques Appliquées I.P.
	MOREAU René	Hydraulique I.P.
	BRUGEL L.	Energétique I.U.T.
	SIBILLE R.	Construction Mécanique I.U.T.
	ARMAND Yves	Chimie I.U.T.
	BOLLIET Louis	Informatique I.U.T.
	KUHN Gérard	Energétique I.U.T.
	GERMAIN J.P.	Construction Mécanique I.U.T.
	CONTE René	Thermodynamique
	JOLY Jean René	Mathématiques Pures
Mme	PIERY Yvette	Biologie Animale
	BENZAKEN Claude	Mathématiques Appliquées

MAITRES DE CONFERENCES ASSOCIES

MM.	SAWCZUK A.	Mécanique des Fluides
	CHEEKE J.	Thermodynamique
	YAMADA O.	Physique du Solide
	NATR Lubomir	B.M.P.V.
	NAVLOK Arch	Physique Industrielle
	SILBER Léo	Radioélectricité
	NOZAKI Akihiro	Mathématiques Appliquées
	RUTLEDGE Joseph	Mathématiques Appliquées
	DONOHU Paul	Physique Générale
	EGGER Kurt	B.M.P.V.

Je tiens à exprimer ma profonde reconnaissance

à

Monsieur le Professeur KUNTZMANN, Directeur de l'Institut de Mathématiques Appliquées de l'Université de Grenoble qui a bien voulu me faire l'honneur de présider le jury,

à

Monsieur le Professeur GASTINEL, qui m'a proposé l'idée de ce travail et qui par ses conseils et ses suggestions en a dirigé l'élaboration. Je l'en remercie vivement,

à

Monsieur LAURENT, Maître de Conférences, qui a eu la bienveillance d'examiner ce travail.

Je tiens aussi à remercier toutes les personnes du Laboratoire qui m'ont apporté leur aide : en particulier Monsieur SIRET pour ses conseils en programmation, Mademoiselle BICAIS qui a apporté le plus grand soin à la réalisation pratique de cette thèse et Monsieur MOUNET pour l'impression.

TABLE DES MATIERES

Page

CHAPITRE I - CONSTRUCTION DE MATRICES TESTS POUR LE CALCUL DES COEFFICIENTS DU POLYNOME CARACTERISTIQUE

II	Procédure Algol	
III	Précision et exemples	7

CHAPITRE II - TRACES DES PUISSANCES SUCCESSIVES D'UNE MATRICE A

I	Introduction	10
II	Existence et détermination d'une matrice de potentiel δA de A	11
III	Structure de la matrice de perturbation $\delta'A$ de $A + \epsilon I$	14
IV	Détermination de paramètres optimum ϵ	17
V	Exemples numériques	31

CHAPITRE III - INVERSION FORMELLE DES RELATIONS DE NEWTON

I	Introduction	25
II	Expression d'une fonction symétrique d'ordre quelconque des racines d'une équation en fonction des sommes de puissances semblables des racines	27
III	Détermination des coefficients d'une équation en fonction des sommes de puissances semblables des racines	33

IV	Résolution de l'équation $1.\lambda_1 + 2.\lambda_2 + \dots + i.\lambda_i = i$	33
V	Procédures Algol	37
VI	Résultats	49

CHAPITRE IV - CALCUL FORMEL DES COEFFICIENTS D'UN POLYNOME EN LISP 1-5

I	Introduction	64
II	Principe	64
III	Mise en oeuvre de l'algorithme	65
IV	Développement et simplification des expressions arithmétiques	68
V	Résultats	74

CHAPITRE V - CALCUL DE COEFFICIENTS DU POLYNOME CARACTERISTIQUE D'UNE MATRICE A

I	Introduction	80
II	Méthode de Leverrier	80
III	Comparaison de cette méthode avec le calcul formel	81

CHAPITRE VI - COEFFICIENTS DU POLYNOME CARACTERISTIQUE ET MATRICES ANTISCALAIRES

I	Introduction	85
II	Détermination de δA sous la forme uv^T , $u, v \in k^n$	86
III	Existence d'une solution	88
IV	Recherche de la solution de norme minimum dans le cas symétrique	89
V	Applications	93

CHAPITRE VII - METHODE DE NEWTON POUR LE PROBLEME DES ELEMENTS
PROPRES D'UNE MATRICE SYMETRIQUE

I	Principe de la méthode	99
II	Problème $\mathcal{P}(n)$	101
III	Problème de type $\mathcal{Q}(n)$	105
IV	Convergence de procédé	108
V	Résolution numérique des problèmes $\mathcal{P}(n)$ et $\mathcal{Q}(n)$.	109
VI	Procédures Algol	112
VII	Résultats et conclusion	113

BIBLIOGRAPHIE

INTRODUCTION

L'idée originale de ce travail, qui me fut proposé par Monsieur N. GASTINEL, était d'étudier une méthode de perturbation pour le problème de la recherche des coefficients du polynôme caractéristique d'une matrice. En fait, de part la difficulté qu'il y a à développer $\det(A+\delta A-\lambda I)$ ou δA est une matrice de perturbation de la matrice $A \in M_{n,n}(\mathbb{R})$, il ne nous a pas été possible d'obtenir des résultats concluants, si ce n'est dans le cas où δA est de la forme uv^T , u et $v \in \mathbb{R}^n$.

Notre premier travail fut de construire des matrices dont on connaisse à l'avance les coefficients du polynôme caractéristique, ce que nous exposons dans le chapitre I. La méthode de Leverrier pour le calcul des coefficients du polynôme caractéristique d'une matrice A faisant intervenir le calcul des traces des puissances successives de la matrice A , nous avons alors étudié de quelle façon perturber la matrice A donnée par une matrice δA pour que le calcul effectif (sur un matériel donné) des traces des puissances successives de $A+\delta A$ nous donne un résultat plus proche du résultat exact que le même calcul effectué sur la matrice A (chapitre II).

Cette méthode de Leverrier utilisant les relations de Newton entre les coefficients d'un polynôme et les sommes des puissances semblables de ses racines, nous avons cherché l'expression de ces coefficients en fonction des sommes des puissances semblables de deux façons différentes :

- en utilisant les formules données par Warring au siècle dernier (ch. III)
- en construisant directement ces formules sur l'ordinateur dont nous disposons, à l'aide du langage LISPl-5 (chapitre IV).

Nous avons alors, dans le chapitre V, comparé les résultats retenus en utilisant d'une part les relations de Newton, d'autre part les formules de Warring.

Dans le chapitre VI nous étudions une perturbation de la forme uv^T ,
 u et $v \in \mathbb{R}^n$, pour le calcul du polynôme caractéristique d'une matrice.

Nous nous sommes ensuite intéressés à des méthodes de perturbations, en ce qui concerne la diagonalisation d'une matrice A , du type suivant : étant donnée $A \in \mathcal{M}_{n,n}(\mathbb{R})$ diagonalisable nous savons qu'il existe une matrice B régulière telle que : $B^{-1}AB = \Delta$ soit diagonale. Notre but était de construire une suite de matrices $\{B_i\}$ telles que :

$$(1) \quad \|B_i^{-1}AB_i\|_* < \|B_{i-1}^{-1}AB_{i-1}^{-1}\|_*,$$

$\|M\|_*$ désignant la norme euclidienne des éléments non diagonaux de la matrice M , la matrice B_i étant déterminée par :

$B_i = B_{i-1} + \delta B_{i-1}$, δB_{i-1} devant être divisée de façon à satisfaire (1).

Nous faisons alors une approximation, au premier ordre en δB_i , pour calculer l'inverse B_i^{-1} de B_i , à savoir :

$$B_i^{-1} = B_{i-1}^{-1} - B_{i-1}^{-1} \delta B_{i-1} B_{i-1}^{-1},$$

mais du fait de cette approximation il ne nous a pas été possible de prouver la convergence du procédé.

Monsieur GASTINEL m'a alors proposé, pour la diagonalisation d'une matrice symétrique, une généralisation de la méthode de Newton pour la recherche de la solution de $f(x) = 0$, qui est exposée dans le chapitre VII.

CHAPITRE - I

CONSTRUCTION DE MATRICES TESTS

Pour étudier une méthode de calcul des coefficients du polynôme caractéristique A, il est intéressant, pour les applications numériques de disposer de matrices dont les coefficients du polynôme caractéristique peuvent être connus simplement. Nous avons alors le choix entre deux solutions :

- ou bien prendre une matrice quelconque dont on peut déterminer les coefficients du polynôme caractéristique par un autre moyen : par exemple en la transmutant en une forme triple diagonale et en calculant alors le polynôme caractéristique de cette forme tridiagonale.
- ou bien construire une matrice dont nous connaissons à priori les coefficients du polynôme caractéristique. C'est ce que nous ferons le plus souvent et nous allons indiquer un moyen de construire de telles matrices.

Soient 2 vecteurs U et V appartenant à R^n et la matrice antisymétrique

$$K = UV^T \quad [1]$$

Alors $K^2 = UV^TUV^T = V^TU(UV^T) = (V^TU)K$. Soit $p = V^TU$.

K est une matrice à polynôme minimal du second degré $K^2 - pK = 0$ si $p \neq 0$, et si $p = 0$ alors $V^TU = 0$: les 2 vecteurs sont orthogonaux et le polynôme minimal de K se réduit à $K^2 = 0$.

Soient a et $b \in \mathbb{R}$ et

$$M = aI + bK.$$

L'inverse M^{-1} de M , si elle existe est du même type que M

$$M^{-1} = a'I + b'K, \quad a' \text{ et } b' \text{ étant déterminés par :}$$

$$M^{-1}M = MM^{-1} = aa'I + [(a+bp)b' + ba']K = I$$

Soit : si $\begin{cases} a \neq 0 \\ a+bp \neq 0 \end{cases}$

$$\begin{cases} a' = 1/a \\ b' = -b/[a(a+bp)]. \end{cases}$$

Soient alors c_1, c_2, \dots, c_n les n nombres qui servent les coefficients du polynôme caractéristique de la matrice A , le coefficient c_0 du terme de degré n étant pris égal à $(-1)^n$. On sait d'après la théorie de la réduction des matrices à leur première forme normale que si une matrice admet des racines simples à son polynôme caractéristique, il est possible de trouver une matrice semblable à celle-ci et de la forme de Frobenius [2].

$$F = \begin{vmatrix} 0 & 0 & & & -c_n \\ 1 & 0 & & & -c_{n-1} \\ 0 & 1 & 0 & & -c_{n-2} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 1 & 0 & & -c_1 \end{vmatrix}$$

Nous allons construire $A = B \mathcal{G} B^{-1}$ ou B est à polynôme minimal du second degré :

$$B = (\beta-1)I + K \text{ avec } K = \sum_{i,j} E_{i,j} = \begin{vmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 1 & & & 1 \\ | & & & & | \\ | & & & & | \\ 1 & & & & 1 \end{vmatrix}$$

On a alors : $B^{-1} = \alpha'I + \beta'K$

avec

$$\alpha' = \frac{1}{\beta-1} \quad \beta' = \frac{-1}{\beta-1+n}$$

En explicitant B , \mathcal{G} et B^{-1} :

$$A = (\beta_0 I + K) (H - \bar{\gamma}) (\alpha' I + \beta' K)$$

en posant : $\beta_0 = \beta - 1$

$$H = \begin{vmatrix} 0 & & & 0 \\ 1 & 0 & & \\ & 1 & 0 & \\ & & \ddots & \ddots \\ 0 & & & 1 & 0 \end{vmatrix} \quad \bar{\gamma} = \begin{vmatrix} 0 & \dots & c_n \\ & & c_{n-1} \\ & & \vdots \\ & & c_1 \end{vmatrix} = \gamma \cdot e_n^T$$

avec γ vecteur de \mathbb{R}^n :

$$\gamma = \begin{vmatrix} c_n \\ c_{n-1} \\ \vdots \\ c_1 \end{vmatrix}, \quad e_n \text{ n}^{\text{ième}} \text{ vecteur de la base fondamentale de } \mathbb{R}^n.$$

Dans ces conditions on a en développant :

$$\begin{aligned}
 A &= [\beta_0 (H - \bar{\gamma}) + KH - K\bar{\gamma}] [\alpha' I + \beta' K] \\
 &= \alpha' \beta_0 (H - \bar{\gamma}) + \beta_0 \beta' (H - \bar{\gamma}) K + \alpha' KH + \beta' KHK - \alpha' K\bar{\gamma} - \beta' K\bar{\gamma} K \\
 &= \alpha' \beta_0 H - \alpha' \beta_0 \bar{\gamma} + \beta_0 \beta' HK - \beta_0 \beta' \bar{\gamma} K + \alpha' KH + \beta' KHK - \alpha' K\bar{\gamma} - \beta' K\bar{\gamma} K.
 \end{aligned}$$

Explicitons les différents termes en développement :

$$\cdot KH = \left(\sum_{i,j} E_{i,j} \right) \left(\sum_{\ell=2}^n E_{\ell, \ell-1} \right) = \sum_{i,j} \left(\sum_{\ell=2}^n E_{i,j} E_{\ell, \ell-1} \right) = \sum_{i=1}^n \sum_{j=2}^n E_{i, j-1}$$

$$\underline{KH = \sum_{i=1}^n \sum_{j=1}^{n-1} E_{i,j}}$$

$$\cdot KH = \sum_{\ell=2}^n E_{\ell, \ell-1} \left(\sum_{i,j} E_{i,j} \right) = \sum_{\ell=2}^n \left(\sum_i \sum_j E_{\ell, \ell-1} \cdot E_{i,j} \right) = \sum_{\ell=2}^n \sum_{j=1}^n E_{\ell, j}$$

$$\underline{KH = \sum_{i=2}^n \sum_{j=1}^n E_{i,j}}$$

$$\cdot KHK = \left(\sum_{i,j} E_{i,j} \right) \left(\sum_{k=2}^n \sum_{\ell=1}^n E_{k\ell} \right) = \sum_{i,j} \left(\sum_{k=2}^n \sum_{\ell=1}^n E_{i,j} E_{k\ell} \right) = (n-1) \sum_i \sum_{\ell} E_{i,\ell}$$

$$\underline{KHK = (n-1) \sum_{i,j} E_{i,j}}$$

$$\cdot K\bar{\gamma} = \sum_{i,j} E_{i,j} \times \gamma \cdot e_n^T = \left(\sum_i \sum_j c_{n+1-j} \cdot e_i \right) e_n^T = S \sum_i E_{i,n}$$

$$\text{avec } S = \sum_{j=1}^n c_{n+1-j}$$

$$\underline{K\bar{\gamma} = S \sum_{i=1}^n E_{i,n}}$$

$$\begin{aligned}
 \bar{\gamma}K &= \sum_{\ell=1}^n c_{n+1-\ell} E_{\ell,n} \cdot \sum_i \sum_j E_{i,j} \\
 &= \sum_{\ell=1}^n c_{n+1-\ell} \left(\sum_{i,j} E_{\ell,n} E_{i,j} \right) \\
 &= \sum_{\ell=1}^n \sum_{j=1}^n c_{n+1-\ell} E_{\ell,j}
 \end{aligned}$$

$$\underline{\bar{\gamma}K} = \sum_i \sum_j c_{n+1-i} E_{i,j}$$

$$K\bar{\gamma}K = S \sum_{i=1}^n E_{i,n} \sum_{k,\ell} E_{k,\ell} = S \sum_{i=1}^n \left(\sum_{k,\ell} E_{i,n} E_{k,\ell} \right)$$

$$\underline{K\bar{\gamma}K} = S \sum_{i=1}^n \sum_{\ell=1}^n E_{i,\ell}$$

D'où l'élément en position i,j de A :

$$\begin{aligned}
 a_{i,j} &= \alpha' \beta_0 \delta_{i,j+1} - \alpha' \beta_0 c_{n+1-i} \delta_{n,j} + \beta_0 \beta' (1 - \delta_{i1}) \\
 &\quad - \beta_0 \beta' c_{n+1-i} + (1 - \delta_{n,j}) \alpha' + (n-1) \beta' - \delta_{n,j} \alpha' S - \beta' S
 \end{aligned}$$

$$\begin{aligned}
 a_{i,j} &= \alpha' [1 - \delta_{n,j} + \beta_0 \delta_{i,j+1} - \beta_0 c_{n+1-i} \delta_{n,j} - S \delta_{n,j}] \\
 &\quad + \beta' [\beta_0 (1 - \delta_{i1}) - \beta_0 c_{n+1-i} + (n-1) - S]
 \end{aligned}$$

et en remplaçant β_0 par $\beta-1$ et en regroupant les termes :

$$\begin{aligned}
 a_{i,j} &= \beta' [(\beta+n-1) - (\beta-1) c_{n+1-i} - S - 1 - \delta_{i1} (\beta-1)] + \\
 &\quad \alpha' [1 + \delta_{i,j+1} (\beta-1) - \delta_{n,j} (c_{n+1-i} (\beta-1) + S + 1)]
 \end{aligned}$$

$$\text{Soit alors : } \begin{cases} \gamma_i = -S - (\beta - 1)c_{n+1-i}^{-1} \\ \beta + n - 1 = \mu \\ \beta - 1 = \lambda \end{cases} \quad \begin{cases} \alpha' = \frac{1}{\lambda} \\ \beta' = -\frac{1}{\lambda\mu} \end{cases}$$

l'élément $a_{i,j}$ de A est alors donné par :

$$a_{i,j} = \beta' [\mu + \gamma_i - \delta_{i1} \lambda] + \alpha' [1 + \delta_{i,j+1} \lambda + \gamma_i \delta_{n,j}]$$

soit :

$$a_{i,j} = \frac{\gamma_i}{\lambda\mu} + \frac{1}{\mu} \delta_{i1} + \delta_{i,j+1} + \frac{\gamma_i}{\lambda} \delta_{n,j}$$

Pour cette matrice les coefficients du polynôme caractéristique sont $c_0 = (-1)^n$, c_1, c_2, \dots, c_n donnés.

Voici une procédure Algol permettant de construire une matrice-test par ce procédé :

procédure MATRICETEST(M,A,N,BETA) ;

entier N ; réel BETA ; tableau M,A ;

commentaire cette procédure forme une matrice M d'ordre N, dont les coefficients du polynôme caractéristique sont connus et rangés dans le tableau A, dans l'ordre des puissances croissantes. β est un paramètre variable différent de 1 et de 1-N ;

début réel G,S,T,U ; entier I,J ;

S := 0.0 ;

pour I := 1 pas 1 jusqua N faire

S := S - A[I] ;

T := BETA - 1 ; U := T + N ;

```

    pour I := 1 pas 1 jusqu'a N faire
  début
    G := S-A[I]*T-1 ;
    pour J := 1 pas 1 jusqu'a N faire
      si I = 1 alors M[I,J] := (1-G/T)/U sinon
        M[I,J] := -G/(T*U) ;
      M[I,N] := M[I,N]+G/T ;
    fin ;
  pour I := 2 pas 1 jusqu'a N faire
    M[I,I-1] := M[I,I-1]+1 ;
  fin PROCEDURE ;

```

Précision obtenue : Remarquons que le nombre d'opérations nécessaire pour obtenir chaque élément de la matrice est très réduit. Il suffit de calculer une fois pour toutes la somme S des coefficients du polynôme caractéristique (non compris le coefficient du terme de plus haut degré) ce qui pour une matrice d'ordre n nécessite n-1 additions. Ensuite chacun des termes s'obtient au bout de 5 additions et 4 multiplications au plus.

De plus β est un paramètre variable et il est possible de choisir sa valeur de façon que les divisions par $\beta-1$ ou $\beta-1+n$ donnent des résultats ayant un nombre fini de décimales. Dans ces conditions les résultats obtenus sont sans erreur ainsi que le prouvent les résultats ci-dessous :

Exemple 1 :

Nous voulons construire une matrice carrée d'ordre 6 ayant pour polynôme caractéristique :

$$\det(A-\lambda I) = \lambda^6 - 1.24\lambda^5 + 3.68\lambda^4 - 2.274\lambda^3 + 6.424\lambda^2 - 12.325\lambda + 24.875$$

Voici la matrice exacte correspondant à $\beta = 5$:

A exacte	0.4795	0.4795	0.4795	0.4795	0.4795	-3.3155
	1.8715	0.8715	0.8715	0.8715	0.8715	-7.8435
	0.2761	1.2761	0.2761	0.2761	0.2761	-2.4849
	1.1459	1.1459	2.1459	1.1459	1.1459	-10.3131
	-0.729	-0.729	-0.729	0.271	-0.729	6.561
	2.991	2.991	2.991	2.991	3.991	-26.919

et voici la matrice calculée par la procédure MATRICETEST :

0.47950000	0.47950000	0.47950000	0.47950000	0.47950000	-3.3155000
1.8715000	0.871 <u>49998</u>	0.871 <u>49998</u>	0.871 <u>49998</u>	0.871 <u>49998</u>	-7.843 <u>4999</u>
0.27609999	1.2761000	0.27609999	0.27609999	0.27609999	-2.4849000
1.1459000	1.1459000	2.1459000	1.1459000	1.1459000	-10.313100
-0.72900000	-0.72900000	-0.72900000	0.27100000	-0.72900000	6.5610000
2.9910000	2.9910000	2.9910000	2.9910000	3.9910000	-26.919000

Les chiffres inexacts sont soulignés.

La norme euclidienne de l'erreur est égale à : $0.28868 \cdot 10^{-6}$.

L'erreur relative vaut : $0.9012 \cdot 10^{-8}$.

Norme du max de l'erreur : $0.23842 \cdot 10^{-6}$.

Exemple 2 :

Matrice carrée d'ordre 8 de polynôme caractéristique :

$$\det(A-\lambda I) = \lambda^8 - 124\lambda^7 + 256\lambda^6 - 78\lambda^5 + 192\lambda^4 - 242\lambda^3 + 144\lambda^2 - 26\lambda + 746.$$

Nous avons pris pour le calcul $\beta = -3$ et le calcul des éléments de la matrice par la procédure MATRICETEST donne un résultat absolument exact : la matrice calculée est égale à la matrice exacte.

Matrice calculée :

-85.0625	-85.0625	-85.0625	-85.0625	-85.0625	-85.0625	-85.0625	256.1875
10.6875	9.6875	9.6875	9.6875	9.6875	9.6875	9.6875	-29.0625
-73.8125	-72.8125	-73.8125	-73.8125	-73.8125	-73.8125	-73.8125	221.4375
-6.3125	-6.3125	-5.3125	-6.3125	-6.3125	-6.3125	-6.3125	18.9375
-114.8125	-114.8125	-114.8125	-113.8125	-114.8125	-114.8125	-114.8125	344.4375
-18.3125	-18.3125	-18.3125	-18.3125	-17.3125	-18.3125	-18.3125	54.9375
-60.8125	-60.8125	-60.8125	-60.8125	-60.8125	-59.8125	-60.8125	182.4375
132.1875	132.1875	132.1875	132.1875	132.1875	132.1875	133.1875	-396.5625

Nous voyons donc qu'il est possible de construire, sans erreur, une matrice dont les coefficients du polynôme caractéristique sont donnés à l'avance.

CHAPITRE - II

TRACES DES PUISSANCES SUCCESSIVES D'UNE MATRICE A.

I - En étudiant la méthode de Leverrier pour le calcul des coefficients du polynôme caractéristique d'une matrice A, nous avons été amenés à étudier le calcul des traces des puissances successives de cette matrice.

Soit donc une matrice $A \in \mathcal{M}_{n,n}(\mathbb{R})$. Nous désirons calculer Trace (A), Trace (A^2), ..., Trace (A^n).

Soit \mathcal{M} l'application qui à une matrice A fait correspondre le vecteur $T \in \mathbb{R}^n$ de composantes :

$$t_i = \text{Trace}(A^i).$$

$$(1) \quad A \xrightarrow{\mathcal{M}} T$$

Du fait des erreurs de calcul dues aux opérations d'arrondi et de troncature au cours du calcul effectif, sur un matériel donné, à l'application \mathcal{M} , il correspond numériquement un algorithme \mathcal{M}^* qui à la matrice A fait correspondre un vecteur T^* .

$$(2) \quad A \xrightarrow{\mathcal{M}^*} T^*$$

Supposons alors qu'il existe une matrice $\delta A \in \mathcal{M}_{n,n}(\mathbb{R})$ telle que l'on ait :

$$(3) \quad A + \delta A \xrightarrow{\mathcal{M}} T^*$$

T^* est alors le résultat exact de l'application \mathcal{M} , appliquée à $A + \delta A$.

Soient alors ϵ réel et H une matrice que nous préciserons ultérieurement. Appliquons alors \mathcal{M}_ϵ^* à la matrice $A+\epsilon H$

$$(4) \quad A+\epsilon H \xrightarrow{\mathcal{M}_\epsilon^*} T_\epsilon^*$$

Nous obtenons alors un vecteur T_ϵ^* que nous pouvons considérer comme le résultat exact de l'application \mathcal{M}_ϵ appliqué à la matrice : $A+\epsilon H+\delta'A$.

$$(5) \quad A+\epsilon H+\delta'A \xrightarrow{\mathcal{M}_\epsilon} T_\epsilon^*$$

Si nous pouvons alors déterminer ϵ de telle sorte que la norme de $\epsilon H+\delta'A$ soit inférieure à la norme de δA , le problème (5) sera plus proche du problème (1) que le problème (3) et par "continuité" de l'algorithme \mathcal{M}_ϵ nous pouvons penser que la solution T_ϵ^* du problème (5) sera plus proche de T que la solution T^* du problème (3).

II - EXISTENCE ET DETERMINATION DE LA MATRICE δA .

$$(a) \quad \begin{array}{l} \text{Soient} \\ \left\{ \begin{array}{l} t_1 = \text{trace}(A) \\ t_2 = \text{trace}(A^2) \\ \vdots \\ t_n = \text{trace}(A^n) \end{array} \right. \end{array} \quad \begin{array}{l} \text{Numériquement nous obtiendrons} \\ \left\{ \begin{array}{l} \tilde{t}_1 \\ \tilde{t}_2 \\ \vdots \\ \tilde{t}_n \end{array} \right. \end{array}$$

Existe-t-il alors une matrice δA telle que l'on ait :

$$(b) \quad \left\{ \begin{array}{l} \tilde{t}_1 = \text{tr}(A+\delta A) \\ \tilde{t}_2 = \text{tr}[(A+\delta A)^2] \\ \vdots \\ \tilde{t}_n = \text{tr}[(A+\delta A)^n] \end{array} \right.$$

Développons $\text{Tr}[(A+\delta A)^k]$.

On a : $(A+\delta A)^k \approx A^k + \sum_{i=0}^{k-1} A^i \delta A A^{k-1-i}$

en négligeant les termes d'ordre supérieur en δA .

$$\begin{aligned} \text{Tr}[(A+\delta A)^k] &= \text{Tr}[A^k] + \text{Tr}\left[\sum_{i=0}^{k-1} A^i \delta A A^{k-1-i}\right] \\ &= \text{Tr}[A^k] + \sum_{i=0}^{k-1} \text{Tr}[A^i \delta A A^{k-1-i}] \end{aligned}$$

et comme $\text{Tr}(A.B) = \text{Tr}(B.A)$ nous aurons :

$$\text{Tr}[(A+\delta A)^k] = \text{Tr}(A^k) + \sum_{i=0}^{k-1} \text{Tr}(A^{k-1} \delta A)$$

Soit

$$\underline{\text{Tr}[(A+\delta A)^k] = \text{Tr}[A^k] + k \cdot \text{Tr}[A^{k-1} \delta A]}$$

Les relations (b) s'écrivent alors :

$$(b') \quad \begin{cases} \tilde{t}_1 = \text{Tr}(A) + \text{Tr}(\delta A) \\ \tilde{t}_2 = \text{Tr}(A^2) + 2\text{Tr}(A\delta A) \\ \tilde{t}_k = \text{Tr}(A^k) + k\text{Tr}(A^{k-1}\delta A) \\ \tilde{t}_n = \text{Tr}(A^n) + n\text{Tr}(A^{n-1}\delta A) \end{cases}$$

De (a) et (b') nous déduisons le système (c) :

$$(c) \quad \begin{cases} \tilde{t}_1 - t_1 = \text{Tr}(\delta A) \\ \tilde{t}_2 - t_2 = 2\text{Tr}(A\delta A) \\ \vdots \\ \tilde{t}_k - t_k = k \cdot \text{Tr}(A^{k-1}\delta A) \\ \tilde{t}_n - t_n = n \cdot \text{Tr}(A^{n-1}\delta A) \end{cases}$$

$\text{Tr}(\delta A), \text{Tr}(A\delta A), \dots, \text{Tr}(A^{n-1}\delta A)$ sont linéaires en les éléments de la matrice δA . Nous disposons ainsi de n relations pour déterminer les n^2 éléments de la matrice δA . Ce problème admet en général une infinité de solutions : pour en obtenir une, nous fixerons $n^2 - n$ coefficients de δA de façon arbitraire et nous utiliserons les relations (c) pour déterminer les n autres.

Soit $\alpha_{i,j}$ l'élément en position (i,j) de δA . Cherchons alors δA sous forme diagonale. $\alpha_{i,j} = 0$ si $i \neq j$ ce qui revient à fixer à zéro $n^2 - n$ paramètres du système (c). Déterminons alors les α_{ii} que nous noterons α_i par la suite afin d'alléger les notations.

Le système s'écrit alors :

$$(c') \quad \begin{cases} \tilde{t}_1 - t_1 = \sum_{i=1}^n \alpha_i \\ \tilde{t}_2 - t_2 = 2 \cdot \sum_{i=1}^n a_{ii}^{(2)} \alpha_i \\ \vdots \\ \tilde{t}_k - t_k = k \cdot \sum_{i=1}^n a_{ii}^{(k)} \alpha_i \\ \vdots \\ \tilde{t}_n - t_n = n \cdot \sum_{i=1}^n a_{ii}^{(n)} \alpha_i \end{cases} \quad \text{où } a_{ii}^{(k)} \text{ représente l'élément en position } (i,i) \text{ de la matrice } A^k.$$

Les α_i sont alors solutions du système linéaire :

$$(I) \quad \begin{bmatrix} 1 & 1 & \dots & \dots & 1 \\ a_{11}^{(2)} & a_{22}^{(2)} & \dots & \dots & a_{nn}^{(2)} \\ \vdots & \vdots & \dots & \dots & \vdots \\ a_{11}^{(n-1)} & a_{22}^{(n-1)} & \dots & \dots & a_{nn}^{(n-1)} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} \tilde{t}_1 - t_1 \\ (\tilde{t}_2 - t_2)/2 \\ (\tilde{t}_3 - t_3)/3 \\ \vdots \\ (\tilde{t}_n - t_n)/n \end{bmatrix}$$

Soit $\boxed{B \cdot \alpha = \delta T}$

Si la matrice B n'est pas singulière, le système (I) admet alors une solution unique et ceci nous permet de déterminer les α_i .

D'où la conclusion :

Le vecteur $T^* \in R^n$ ayant pour composantes les traces des puissances successives (de 1 à n) d'une matrice $A \in M_{n,n}(R)$ peut être considéré comme le vecteur T^* ayant pour composantes les traces exactes des puissances successives (de 1 à n) de la matrice $(A+\delta A)$, δA étant une matrice déterminée comme ci-dessus.

$$A \xrightarrow{H} T^* \xrightarrow{A+\delta A}$$

III - STRUCTURE DE LA MATRICE $\delta'A$

Soit maintenant $\epsilon \in R$ et $H \in M_{n,n}(R)$. Formons $A+\epsilon H$ et

Soient

$$(a) \quad \begin{cases} t'_1 = \text{Tr}(A+\epsilon H) \\ t'_2 = \text{Tr}[(A+\epsilon H)^2] \\ t'_n = \text{Tr}[(A+\epsilon H)^n] \end{cases} \quad \text{Numériquement nous obtiendrons} \quad \begin{cases} \tilde{t}'_1 \\ \tilde{t}'_2 \\ \tilde{t}'_n \end{cases}$$

D'après l'étude précédente nous savons qu'il existe une matrice $\delta'A$ telle que l'on ait exactement :

$$(b) \quad \begin{cases} \tilde{t}'_1 = \text{Tr}(A+\epsilon H+\delta'A) \\ \tilde{t}'_2 = \text{Tr}[(A+\epsilon H+\delta'A)^2] \\ \tilde{t}'_n = \text{Tr}[(A+\epsilon H+\delta'A)^n] \end{cases}$$

En développant $\text{Tr}[(A+\epsilon H+\delta'A)^k]$ nous savons que :

$$\text{Tr}[(A+\epsilon H+\delta'A)^k] = \text{Tr}[(A+\epsilon H)^k] + k \cdot \text{Tr}[(A+\epsilon H)^{k-1} \delta'A]$$

d'où les relations (c) :

$$(c) \quad \begin{cases} \tilde{t}'_1 - t'_1 = \text{Tr}(\delta'A) \\ \tilde{t}'_2 - t'_2 = 2\text{Tr}[(A+\epsilon H)\delta'A] \\ \tilde{t}'_k - t'_k = k\text{Tr}[(A+\epsilon H)^{k-1} \delta'A] \\ \tilde{t}'_n - t'_n = n\text{Tr}[(A+\epsilon H)^{n-1} \delta'A] \end{cases}$$

Développons $\text{Tr}[(A+\epsilon H)^k \delta'A]$

$$\begin{aligned} \text{Tr}[(A+\epsilon H)^k \delta'A] &= \text{Tr}\left[A^k + \epsilon \sum_{i=0}^{k-1} A^i H A^{k-1-i}\right] \delta'A \quad \text{au premier ordre en } \epsilon : \\ &= \text{Tr}[A^k \delta'A] + \epsilon \text{Tr}\left[\sum_{i=0}^{k-1} A^i H A^{k-1-i} \delta'A\right] \end{aligned}$$

Pour simplifier cette expression nous allons particulariser H et dans tout ce qui suit nous poserons H = I matrice unité.

$$\text{Alors : } \text{Tr}[(A+\epsilon I)^k \delta'A] = \text{Tr}[A^k \delta'A] + k \epsilon \text{Tr}(A^{k-1} \delta'A)$$

Les relations (c) s'écrivent alors :

$$(c') \quad \begin{cases} \tilde{t}'_1 - t'_1 = \text{Tr}(\delta'A) \\ \tilde{t}'_2 - t'_2 = 2\text{Tr}(A\delta'A) + 2\epsilon \text{Tr}(\delta'A) \\ \tilde{t}'_k - t'_k = k\text{Tr}(A^{k-1} \delta'A) + k(k-1) \epsilon \text{Tr}(A^{k-2} \delta'A) \\ \tilde{t}'_n - t'_n = n\text{Tr}(A^{n-1} \delta'A) + n(n-1) \epsilon \text{Tr}(A^{n-2} \delta'A) \end{cases}$$

B' s'écrit en effet :

$$B' = (I + \epsilon E_{21})(I + 2\epsilon E_{32}) \dots (I + (k-1)\epsilon E_{k,k-1}) \dots (I + (n-1)\epsilon E_{n,n-1})B.$$

où $E_{i,j}$, matrice de la base de l'espace à n^2 dimensions des matrices carrées sur K , est défini par :

$$E_{i,j} = \begin{array}{c} \begin{array}{c} i^{\circ} \rightarrow \\ \left| \begin{array}{ccc} 0 & \vdots & 0 \\ \hline & 1 & \\ \hline 0 & \vdots & 0 \end{array} \right| \end{array} \\ \begin{array}{c} j^{\circ} \\ \vdots \\ 1 \\ \vdots \\ 0 \end{array} \end{array}$$

$$B' = \prod_{k=1}^{n-1} (I + k\epsilon E_{k+1,k}).B$$

Soit $B' = P.B$ avec $P = \prod_{k=1}^{n-1} (I + k\epsilon E_{k+1,k})$.
d'où

$$P.B.\alpha' = \delta'T.$$

La matrice P est inversible ($\det(P)=1$) d'où, si B est elle-même inversible :

$$\alpha' = B^{-1}.P^{-1}.\delta'T.$$

IV - DETERMINATION DU PARAMETRE ϵ

Nous désirons déterminer ϵ de telle sorte que l'on ait :

$$(a) \quad \|\delta'A + \epsilon I\| < \|\delta A\|$$

Remarquons que pour la norme euclidienne ou la norme du max, l'inégalité (a) est équivalente à :

$$\|\alpha' + \epsilon X\| < \|\alpha\| \quad \text{ou } X \text{ est le vecteur de } \mathbb{R}^n \text{ défini par } X = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

Est-il possible de choisir ϵ de telle sorte que :

$$(b) \quad \|B^{-1}P^{-1}(\epsilon)\delta'T + \epsilon X\| < \|B^{-1}\delta T\|.$$

Il paraît naturel de penser que le paramètre ϵ dépendra non seulement de la norme des vecteurs qui figurent dans l'inégalité (b) mais aussi de leur orientation dans l'espace. Aussi allons-nous essayer de mettre en évidence une relation entre les vecteurs δT et $\delta'T$.

a) Relation entre δT et $\delta'T$.

Soit T le vecteur des traces des puissances successives de A (de 1 à n),
et T' le vecteur des traces des puissances successives de $A + \epsilon I$ (de 1 à n).

Soient $(T)_i = t_i$ et $(T')_i = t'_i$.

Les composantes de T' et T vérifient les relations :

$$\begin{cases} t'_1 = t_1 + n\epsilon \\ t'_2 = t_2 + 2\epsilon t_1 + n\epsilon^2 \\ \vdots \\ t'_n = t_n + n\epsilon t_{n-1} + \dots + n\epsilon^n \end{cases}$$

ce que nous pouvons écrire :

Nous voulons déterminer ϵ de sorte que :

$$\|\alpha' + \epsilon X\| < \|\alpha\|$$

soit $\|B^{-1}P^{-1}\delta'T + \epsilon X\| < \|B^{-1}\delta T\|$

$$P^{-1}(\epsilon)\delta'T = P^{-1}(\epsilon)\delta T + \epsilon P^{-1}(\epsilon)K\delta T$$

$$= (P^{-1}(\epsilon) + \epsilon P^{-1}(\epsilon)K)\delta T$$

$$= (I + \epsilon J)\delta T \text{ au premier ordre en } \epsilon$$

avec

$$J = \sum_{i=1}^{n-1} E_{i+1,i}$$

D'où $B^{-1}P^{-1}\delta'T = B^{-1}(I + \epsilon J)\delta T$ et ϵ doit être tel que :

$$\|B^{-1}\delta T + \epsilon B^{-1}J\delta T + \epsilon X\| < \|B^{-1}\delta T\|.$$

soit

$$\|B^{-1}\delta T + \epsilon(B^{-1}J\delta T + X)\| < \|B^{-1}\delta T\|.$$

soit

$$u = B^{-1}J\delta T + X$$

$$\begin{aligned} \|B^{-1}\delta T + \epsilon u\|^2 &= (B^{-1}\delta T + \epsilon u)^T (B^{-1}\delta T + \epsilon u) \\ &= (\delta T B^{-1})^T (B^{-1}\delta T) + 2\epsilon u^T B^{-1}\delta T + \epsilon^2 u^T u \\ &= \|B^{-1}\delta T\|^2 + 2\epsilon u^T B^{-1}\delta T + \epsilon^2 \|u\|^2. \end{aligned}$$

Ce trinôme en ϵ passe par un minimum pour :

$$\epsilon = - \frac{u^T B^{-1}\delta T}{\|u\|^2}$$

et ce minimum vaut alors :

$$\|B^{-1}\delta T\|^2 - \frac{(u^T B^{-1}\delta T)^2}{\|u\|^2}, \text{ qui est bien inférieur à } \|B^{-1}\delta T\|^2.$$

En conclusion :

Etant donné une matrice $A \in \mathcal{M}_{n,n}(\mathbb{R})$, soient :

T le vecteur des traces des puissances successives (de 1 à n) de A ,

T^* le vecteur calculé : T^* est le vecteur exact de $A+\delta A$.

Soient $\varepsilon \in \mathbb{R}$ et $A+\varepsilon I$,

$T(\varepsilon)$ le vecteur des traces des puissances successives de $A+\varepsilon I$,

$T^*(\varepsilon)$ le vecteur calculé : $T^*(\varepsilon)$ est le vecteur exact de $A+\varepsilon I+\delta'A$.

Il est possible de déterminer ε de telle sorte que $A+\varepsilon I+\delta'A$ soit plus proche, en norme, de A que ne l'est $A+\delta A$, ce qui revient à dire que $T^*(\varepsilon)$ est solution d'un problème plus "proche" du problème posé que ne l'est T^* .

V - EXEMPLES NUMERIQUES

Pour justifier ceci nous avons traité un certain nombre d'exemples numériques. Pour cela nous avons construit des matrices dont nous connaissions les valeurs propres, donc les traces exactes et nous avons utilisé les résultats ci-dessus. Ayant calculé les traces de ces matrices nous avons déterminé le paramètre ε et nous avons ensuite recalculé les traces de $A+\varepsilon I$. Voici sur 3 exemples les résultats obtenus, le matériel utilisé étant l'ordinateur IBM 360-67 de l'I.M.A.G.

Exemple 1

Soit la matrice carrée d'ordre 6 :

$$A = \begin{bmatrix} -2.6364 & -25.636 & 10.364 & 18.364 & -17.636 & 16.364 \\ -5.6364 & -15.636 & 8.364 & 16.364 & -19.636 & 14.364 \\ -2.3636 & -24.364 & 5.636 & 19.636 & -16.364 & 17.636 \\ -1.6364 & -23.636 & 12.364 & 11.364 & -15.636 & 18.364 \\ -4.909 & -26.909 & 9.091 & 17.091 & -10.909 & 15.091 \\ -1.8182 & -23.812 & 12.182 & 20.182 & -15.818 & 9.1818 \end{bmatrix}$$

Voici les traces exactes de A, les traces calculées et celle de $A+\epsilon I$.

Traces exactes	Traces calculées	Traces de $A+I$
-4	-4.000019	-4.000006
426	425.9970	425.9968
296	295.9885	295.9948
42690	42689.43	42689.43
114776	114771.6	114773.6
4826226	4826113	4826127

Le paramètre ϵ vaut $\epsilon = 3.177714 \cdot 10^{-6}$.

La norme euclidienne de l'erreur sur A est de : 113.0837

celle sur $A+\epsilon I$ est de : 99.03.

Soit en erreur relative : sur A : $2.342354 \cdot 10^{-5}$

sur $A+\epsilon I$: $2.051297 \cdot 10^{-5}$

Exemple 2

Soit A matrice carrée d'ordre 8.

1.333	17.33	1.333	-38.6	-0.67	25.33	15.33	-18.67
-1.333	12.67	2.667	-37.3	+0.67	26.67	16.67	-17.33
-2.4	17.6	3.6	-38.4	-0.4	25.6	15.6	-18.4
-5.06	14.93	-1.06	-19.07	-3.07	22.93	12.93	-21.06
-2.53	17.47	1.467	-38.53	2.467	25.46	15.46	-18.53
-0.8	19.2	3.2	-36.8	1.2	17.2	17.2	-16.8
-1.467	18.53	2.53	-37.47	0.533	26.53	11.53	-17.47
3.73	16.26	0.266	-39.73	-1.733	24.26	14.26	-7.73

Traces exactes	Traces calculées	Traces de $A+\epsilon I$
2.200000 + 01	2.200000 + 01	2.199996 + 01
8.180000 + 02	8.179943 + 02	8.179948 + 02
1.113400 + 04	1.113389 + 04	1.113385 + 04
2.672670 + 05	2.672619 + 05	2.672618 + 05
5.292862 + 06	5.292753 + 06	5.292753 + 06
1.174331 + 08	1.174304 + 08	1.174304 + 08
2.519858 + 09	2.519792 + 09	2.519794 + 09
5.540820 + 10	5.540634 + 10	5.540662 + 10

Le paramètre ϵ vaut $\epsilon = 4.862622 - 06$

La norme euclidienne de l'erreur sur A est de : 1.860767 + 06

sur $A+\epsilon I$: 1.578239 + 06

soit en erreur relative

sur A : 0.3348786 - 04

sur $A+\epsilon I$: 0.2843674 - 04.

Exemple 3 : Soit la matrice carrée (6,6)

A =

1.95	0.55	-1.85	2.65	2.35	1.15
-0.2	3.6	-2.0	2.5	-2.5	0.1
0.4	0.1	-3.0	3.1	-1.9	1.6
-0.725	-0.125	-2.525	9.375	-3.025	0.475
0.525	1.125	-1.275	3.225	-4.375	1.725
-0.35	0.25	-2.15	2.35	-2.65	5.25

Traces exactes	Traces calculées	Traces de $A+\epsilon I$
12.8	12.8	12.8
97.68	97.67991	97.67994
509.5031	509.5029	509.5031
3546.572	3546.567	3546.568
24077.44	24077.39	24077.39
172925.8	172925.3	172925.3

Le paramètre ϵ vaut : $\epsilon = 3.180229 \cdot 10^{-7}$

Norme euclidienne de l'erreur sur A : $5.641644 \cdot 10^{-1}$

Norme euclidienne de l'erreur sur $A+I$: $5.018716 \cdot 10^{-1}$

soit en erreur relative sur A : $3.230620 \cdot 10^{-6}$

sur $A+\epsilon I$: $2.873989 \cdot 10^{-6}$

CHAPITRE - III

INVERSION FORMELLE DES RELATIONS DE NEWTON

I - INTRODUCTION

Etant donné un polynôme $P_n(x)$ de degré n

$$P_n(x) = a_0 x^n + a_1 x^{n-1} + \dots + a_{n-1} x + a_n$$

nous nous proposons de déterminer les coefficients a_0, a_1, \dots, a_n de ce polynôme lorsqu'on en connaît soit les racines $\xi_1, \xi_2, \dots, \xi_n$, soit les sommes des puissances semblables de ces racines, c'est-à-dire les quantités :

$$S_k = \sum_{j=1}^n \xi_j^k \quad k = 1, \dots, n.$$

Les coefficients a_i ($i = 0, \dots, n$) et les sommes S_k ($k = 1, \dots, n$) satisfont aux relations (établies par Newton) [4] :

$$(1.1) \quad \begin{cases} 1. a_1 + a_0 \cdot S_1 = 0 \\ 2. a_2 + a_1 \cdot S_1 + a_0 \cdot S_2 = 0 \\ \vdots \\ k. a_k + a_{k-1} \cdot S_1 + a_{k-2} \cdot S_2 + \dots + a_0 \cdot S_k = 0 \\ \vdots \\ n. a_n + a_{n-1} \cdot S_1 + \dots + a_0 \cdot S_n = 0 \end{cases}$$

Ayant fixé la valeur de a_0 , la résolution numérique de ces relations est en fait celle d'un système linéaire à matrice de premier membre triangulaire inférieure qui s'écrit :

II - EXPRESSION D'UNE FONCTION SYMETRIQUE D'ORDRE QUELCONQUE DES RACINES D'UNE EQUATION EN FONCTION DES SOMMES DE PUISSANCES SEMBLABLES DES RACINES.

La démonstration que nous donnons ici est tirée du Cours d'Algèbre Supérieure de Monsieur SERRET (1877) [5] et nous l'exposons car elle fait apparaître un algorithme permettant d'utiliser les expressions obtenues.

Soient x_1, x_2, \dots, x_m les m racines d'une équation de degré m et $\alpha_1, \alpha_2, \dots, \alpha_n$ des entiers positifs ou négatifs.

S_{α_1} représentera la somme des puissances de degré α_1 de toutes les racines et le symbole

$$\sigma(\alpha_1, \alpha_2, \dots, \alpha_n) = \sum x_1^{\alpha_1} x_2^{\alpha_2} \dots x_i^{\alpha_i}$$
 désignera la fonction symétrique

d'ordre i , dont tous les termes se déduisent de $x_1^{\alpha_1} x_2^{\alpha_2} \dots x_i^{\alpha_i}$ en faisant toutes les permutations possibles des racines.

Supposons d'abord que tous les exposants α soient inégaux. On aura alors :

$$(1.2) \quad \sigma(\alpha_1, \alpha_2, \dots, \alpha_i) = S_{\alpha_1} \sigma(\alpha_1, \alpha_2, \dots, \alpha_{i-1}) - \sigma(\alpha_1 + \alpha_i, \alpha_2, \dots, \alpha_{i-1}) - \sigma(\alpha_1, \alpha_2 + \alpha_i, \dots, \alpha_{i-1}) - \dots - \sigma(\alpha_1, \alpha_2, \dots, \alpha_{i-1} + \alpha_i)$$

Cette formule permet de calculer les fonctions symétriques d'ordre i connaissant celles de l'ordre $i-1$. On en déduit :

$$(2.2) \quad \left| \begin{array}{l} \sigma(\alpha_1, \alpha_2) = S_{\alpha_1} S_{\alpha_2} - S_{\alpha_1 + \alpha_2} \\ \sigma(\alpha_1, \alpha_2, \alpha_3) = S_{\alpha_1} S_{\alpha_2} S_{\alpha_3} - (S_{\alpha_1} S_{\alpha_2 + \alpha_3} + S_{\alpha_2} S_{\alpha_1 + \alpha_3} + S_{\alpha_3} S_{\alpha_1 + \alpha_2}) + 2S_{\alpha_1 + \alpha_2 + \alpha_3} \\ \text{etc...} \end{array} \right.$$

On peut écrire ces formules d'une manière plus abrégée et trouver leur loi de formation de la façon suivante :

Partageons les i indices $\alpha_1, \alpha_2, \dots, \alpha_i$ en divers groupes.

Soient :

λ_1 le nombre de groupes contenant un seul indice α_i ,

λ_2 le nombre de groupes contenant deux indices,

\vdots

λ_i le nombre de groupes contenant i indices.

On aura : $\lambda_1 + 2\lambda_2 + 3\lambda_3 + \dots + i\lambda_i = i$;

on voit que λ_i doit être égal à zéro ou à l'unité et si $\lambda_i = 1$ tous les autres sont nuls.

Ajoutons alors entre eux les indices α de chaque groupe et désignons par C_1, C_2, \dots, C_μ les sommes obtenues. Enfin considérons le produit $S_{C_1} S_{C_2} \dots S_{C_\mu}$: faisons toutes les permutations possibles d'indices $\alpha_1, \alpha_2, \dots, \alpha_i$ qui font acquérir à ce produit toutes les valeurs distinctes dont il est susceptible ; ajoutons tous les résultats et désignons par $T(\lambda_1, \lambda_2, \dots, \lambda_i)$ la somme obtenue qui sera une fonction symétrique des indices α .

Avec ces notations les formules (2.2) deviennent :

$$(3.2) \quad \left\{ \begin{aligned} \sigma(\alpha_1, \alpha_2) &= T(2, 0) - T(0, 1) \\ \sigma(\alpha_1, \alpha_2, \alpha_3) &= T(3, 0, 0) - T(1, 1, 0) + 2T(0, 0, 1) \\ \sigma(\alpha_1, \alpha_2, \alpha_3, \alpha_4) &= T(4, 0, 0, 0) - T(2, 1, 0, 0) + 2T(1, 0, 1, 0) - 2.3T(0, 0, 0, 1) \end{aligned} \right.$$

et de façon générale :

$$(4.2) \quad \left| \sigma(\alpha_1, \alpha_2, \dots, \alpha_i) = \Sigma (-1)^{i-\lambda_1-\lambda_2-\dots-\lambda_i} \frac{\lambda_1!}{(1!)^{\lambda_1}} \frac{\lambda_2!}{(2!)^{\lambda_2}} \dots \frac{\lambda_i!}{((i-1)!)^{\lambda_i}} T(\lambda_1, \lambda_2, \dots, \lambda_i) \right.$$

le signe Σ du second membre s'étendant à tous les $\lambda_1, \lambda_2, \dots, \lambda_i$ satisfaisant à la condition : $1.\lambda_1 + 2.\lambda_2 + \dots + i.\lambda_i = i$.

Cette formule (4.2), vraie pour $i = 2, 3, 4$ d'après les formules (3.2) se démontre de façon générale en montrant qu'elle est vraie pour $i = k+1$, lorsqu'elle l'est pour $i = k$.

Supposons que l'on ait :

$$(5.2) \quad \sigma(\alpha_1, \alpha_2, \dots, \alpha_k) = \sum (-1)^{k-\lambda_1-\dots-\lambda_k} \frac{\lambda_2^{\lambda_2}}{(2!)} \frac{\lambda_3^{\lambda_3}}{\dots((k-1)!)} \dots \lambda_k^{\lambda_k} T(\lambda_1, \lambda_2, \dots, \lambda_k)$$

la somme portant sur toutes les valeurs des entiers λ telles que

$$\lambda_1 + 2\lambda_2 + \dots + k \cdot \lambda_k = k.$$

L'expression $\sigma(\alpha_1, \alpha_2, \dots, \alpha_{k+1})$ sera formée de termes tels que $T(\lambda_1, \lambda_2, \dots, \lambda_{k+1})$ ou l'on aura : $1 \cdot \lambda_1 + 2 \cdot \lambda_2 + \dots + (k+1) \lambda_{k+1} = k+1$.

Cherchons les coefficients qui multiplient ces différents termes.

Supposons d'abord que λ_1 ne soit pas nul ce qui exige que λ_{k+1} le soit ; on aura : $(\lambda_1 - 1) + 2 \cdot \lambda_2 + \dots + k \cdot \lambda_k = k$.

* Le terme $T(\lambda_1, \lambda_2, \dots, \lambda_k, 0)$ que nous considérons provient en partie (formule (1.2)) de la multiplication de $S_{\alpha_{k+1}}$ par $\sigma(\alpha_1, \alpha_2, \dots, \alpha_k)$; comme les termes de ce produit ne peuvent évidemment se réduire avec ceux qui proviennent du changement de α_1 en α_{k+1} ou de α_2 en $\alpha_2 + \alpha_{k+1}$ etc... Il est clair que le coefficient de $T(\lambda_1, \lambda_2, \dots, \lambda_k, 0)$ dans $\sigma(\alpha_1, \dots, \alpha_{k+1})$ sera égal au coefficient de $T(\lambda_1 - 1, \dots, \lambda_k)$ dans $\sigma(\alpha_1, \alpha_2, \dots, \alpha_k)$ c'est-à-dire égal à :

$$(-1)^{k+1-\lambda_1-\lambda_2-\dots-\lambda_k} \frac{\lambda_2^{\lambda_2}}{(1!)} \frac{\lambda_3^{\lambda_3}}{(2!)} \dots ((k-1)!)^{\lambda_k},$$

résultat qui est conforme à celui déduit de la formule (4.2) si l'on y fait $i = (k+1)$.

* Considérons maintenant le terme en $T(0, \lambda_2, \dots, \lambda_g, \lambda_{g+1}, \dots, \lambda_k, 0)$ dans $\sigma(\alpha_1, \alpha_2, \dots, \alpha_{k+1})$. Ce terme provient tout entier (formule (1.2)) des résultats que l'on obtient en chargeant dans $-\sigma(\alpha_1, \alpha_2, \dots, \alpha_k)$ α_1 en $\alpha_1 + \alpha_{k+1}$ ou α_2 en $\alpha_2 + \alpha_{k+1}$ etc... Les termes de l'expression de $-\sigma(\alpha_1, \alpha_2, \dots, \alpha_k)$ qui concourent ainsi à former le terme que nous considérons sont évidemment de la

forme $T(0, \lambda_2, \dots, \lambda_{g+1}, \lambda_{g+1}^{-1}, \dots, \lambda_k)$ où g peut avoir toutes les valeurs telles que λ_{g+1}^{-1} ne soit pas nul, et le coefficient correspondant sera d'après (3.2) :

$$(-1)^{k+1-\lambda_2-\lambda_3-\dots-\lambda_k} \frac{\lambda_2^{\lambda_2}}{(1!)} \frac{\lambda_3^{\lambda_3}}{(2!)} \dots \frac{\lambda_{g+1}^{\lambda_{g+1}}}{((g-1)!)} \frac{\lambda_{g+1}^{-1}}{(g!)} \dots \frac{\lambda_k^{\lambda_k}}{((k-1)!)}$$

Or chacun des termes de $T(0, \lambda_2, \dots, \lambda_{g+1}, \lambda_{g+1}^{-1}, \dots, \lambda_k)$ contient d'après sa définition même un ou plusieurs facteurs de la forme $S_{\alpha_1+\alpha_2+\dots}$, où le nombre d'indices est égal à g ; si dans les facteurs de ce genre on remplace successivement α_1 par $\alpha_1+\alpha_{k+1}$, puis α_2 par $\alpha_2+\alpha_{k+1}$ puis etc... et qu'on réunisse ensuite tous les résultats, chaque terme sera répété g fois dans la somme. Il s'ensuit que le terme en $T(0, \lambda_2, \dots, \lambda_{g+1}, \lambda_{g+1}^{-1}, \dots, \lambda_k)$ donnera dans $\sigma(\alpha_1, \alpha_2, \dots, \alpha_{k+1})$ une partie des termes contenus dans l'expression $T(0, \lambda_2, \dots, \lambda_g, \lambda_{g+1}, \dots, \lambda_k)$ et que ceux-ci auront pour coefficient :

$$g(-1)^{k+1-\lambda_2-\dots-\lambda_k} \frac{\lambda_2^{\lambda_2}}{(1!)} \dots \frac{\lambda_{g+1}^{\lambda_{g+1}}}{((g-1)!)} \frac{\lambda_{g+1}^{-1}}{(g!)} \dots \frac{\lambda_k^{\lambda_k}}{((k-1)!)}$$

et comme $g \cdot ((g-1)!) = g!$

$$(-1)^{k+1-\lambda_2-\dots-\lambda_k} \frac{\lambda_2^{\lambda_2}}{(1!)} \dots \frac{\lambda_{g+1}^{\lambda_{g+1}}}{((g-1)!)} \frac{\lambda_{g+1}^{-1}}{(g!)} \dots \frac{\lambda_k^{\lambda_k}}{((k-1)!)}$$

Les termes qui peuvent naître des diverses valeurs dont g est susceptible ne pouvant se réduire entre eux, le coefficient de $T(0, \lambda_2, \dots, \lambda_k, 0)$ dans $\sigma(\lambda_1, \lambda_2, \dots, \lambda_{k+1})$ sera :

$$(-1)^{k+1-\lambda_2-\dots-\lambda_k} \frac{\lambda_2^{\lambda_2}}{(1!)} \frac{\lambda_3^{\lambda_3}}{(2!)} \dots \frac{\lambda_k^{\lambda_k}}{((k-1)!)} \text{ résultat conforme à celui}$$

déduit de la formule (4.2) en y faisant $i = k+1$.

* Considérons le terme $T(0, 0, 0, \dots, 0, 1)$ dans $\sigma(\alpha_1, \alpha_2, \dots, \alpha_{k+1})$.

Il se forme au moyen du seul terme $(-1)^k (k-1)! T(0, 0, \dots, 0, 1)$ dans $-\sigma(\alpha_1, \alpha_2, \dots, \alpha_k)$. Il faut effectivement ajouter pour cela, successivement α_{k+1} à chacun des indices qui entrent dans ce terme et réunir tous les k résultats

qui sont évidemment égaux entre eux. Le terme considéré a alors pour valeur :

$$(-1)^k k! T(0,0,\dots,0,1) \text{ d'ou la formule (4.2.)}$$

NOMBRE N DE TERMES CONTENUS DANS LA FONCTION $T(\lambda_1, \lambda_2, \dots, \lambda_i)$

Chacun de ces termes correspond à une certaine distribution des indices $\alpha_1, \alpha_2, \dots, \alpha_i$ en $\lambda_1 + \lambda_2 + \dots + \lambda_i$ groupes, λ_k désignant le nombre de groupes contenant k indices α .

Ecrivons sur une même ligne tous les indices α , de manière que ceux qui appartiennent à un même groupe se trouvent placés les uns à côté des autres, en commençant par les groupes d'une seule lettre puis ceux de 2 lettres etc,...

On pourra former de cette façon N permutations ou arrangements des i indices qui correspondront effectivement aux N termes de T. Si maintenant on fait dans chacun de ces N arrangements toutes les permutations possibles des indices qui composent chaque groupe, sans altérer l'ordre des groupes et sans faire passer aucun indice d'un groupe à un autre, le nombre total d'arrangements que l'on obtiendra sera :

$$N(1!)^{\lambda_1} (2!)^{\lambda_2} \dots ((i-1)!)^{\lambda_{i-1}} (i!)^{\lambda_i}$$

Enfin si dans chacun des arrangements ainsi formés on permute entre eux de toutes les façons possibles, d'abord les λ_1 groupes qui contiennent un indice, puis les λ_2 groupes qui contiennent 2 indices etc... sans altérer l'ordre des indices qui composent un même groupe, le nombre de tous les arrangements obtenus sera :

$$N(1!)^{\lambda_1} (2!)^{\lambda_2} \dots ((i-1)!)^{\lambda_{i-1}} (i!)^{\lambda_i} \lambda_1! \lambda_2! \dots \lambda_i!$$

En opérant ainsi on a formé toutes les $i!$ permutations des i indices sans en omettre une seule ou en rajouter aucune.

D'où :

$$N = i! / (1!)^{\lambda_1} (2!)^{\lambda_2} \dots (i!)^{\lambda_i} \lambda_1! \lambda_2! \dots \lambda_i!$$

Cas particuliers :

* La formule (4.2) suppose que les i indices $\alpha_1, \alpha_2, \dots, \alpha_i$ sont inégaux. Supposons qu'il y en ait μ_1 égaux à α_1, μ_2 égaux à $\alpha_2, \dots, \alpha_k$ égaux à α_k ; le second membre de la formule (4.2) doit être divisé alors par :

$$\mu_1! \mu_2! \dots \mu_k!$$

* Si on suppose que les i indices soient égaux à un même nombre α , on devra diviser le second membre de la formule (4.2) par $i!$; d'ailleurs dans chacun des N termes de $T(\lambda_1, \lambda_2, \dots, \lambda_i)$ il ne reste plus que

$S_{\alpha}^{\lambda_1} S_{2\alpha}^{\lambda_2} \dots S_{i\alpha}^{\lambda_i}$ donc cette fonction a pour valeur

$$i! / ((1!)^{\lambda_1} (2!)^{\lambda_2} \dots (i!)^{\lambda_i}) S_{\alpha}^{\lambda_1} S_{2\alpha}^{\lambda_2} \dots S_{i\alpha}^{\lambda_i}$$

la formule (4.2) devient alors :

(6.2.) $\sigma(\alpha, \dots, \alpha) = \Sigma(x_1, x_2 \dots x_i)^{\alpha} =$

$$\frac{\Sigma(-1)^{i-\lambda_1-\lambda_2-\dots-\lambda_i}}{1^{\lambda_1} 2^{\lambda_2} \dots i^{\lambda_i} \lambda_1! \lambda_2! \dots \lambda_i!} S_{\alpha}^{\lambda_1} S_{2\alpha}^{\lambda_2} \dots S_{i\alpha}^{\lambda_i}, \text{ le 2ème signe}$$

Σ s'étendant à toutes les valeurs positives ou nulles, entières de $\lambda_1, \lambda_2, \dots, \lambda_i$ susceptibles de vérifier :

$$\underline{1.\lambda_1 + 2.\lambda_2 + \dots + i.\lambda_i = i.}$$

III - DETERMINATION DES COEFFICIENTS D'UNE EQUATION EN FONCTION DES SOMMES DE PUISSANCES SEMBLABLES DES RACINES

Soit l'équation $x^m + p_1 x^{m-1} + \dots + p_{m-1} x + p_m = 0$ et soient x_1, x_2, \dots, x_m ses racines.

On a : $p_i = (-1)^i \Sigma x_1 x_2 \dots x_i$ soit en appliquant la formule (6.2) et en y faisant $\alpha = 1$:

$$(1.3) \quad p_i = \Sigma \frac{(-1)^{\lambda_1 + \lambda_2 + \dots + \lambda_i}}{1^{\lambda_1} \cdot 2^{\lambda_2} \cdot \dots \cdot i^{\lambda_i} \lambda_1! \lambda_2! \dots \lambda_i!} s_1^{\lambda_1} s_2^{\lambda_2} \dots s_i^{\lambda_i}$$

la somme Σ s'étendant à tous les i-uples $(\lambda_1, \lambda_2, \dots, \lambda_i)$ satisfaisant à :

$$(2.3) \quad \underline{1.\lambda_1 + 2.\lambda_2 + \dots + i.\lambda_i = i}$$

Afin d'utiliser numériquement la formule (1.3) nous devons résoudre l'équation (2.3) pour les valeurs de i allant de 2 jusqu'à m.

IV - RESOLUTION DE L'EQUATION $1.\lambda_1 + 2.\lambda_2 + \dots + n.\lambda_n = n$

Nous appellerons problème d'ordre n le problème qui consiste à rechercher tous les n-uples $(\lambda_1, \lambda_2, \dots, \lambda_n)$ d'entiers positifs ou nuls satisfaisant à : $1.\lambda_1 + 2.\lambda_2 + \dots + n.\lambda_n = n$ et nous le noterons P_n .

Dans la recherche des solutions du problème P_n , nous supposons connues les solutions des problèmes $P_{n-1}, P_{n-2}, \dots, P_2$, et soit :

$$R_0 : 1.\lambda_1 + 2.\lambda_2 + \dots + n.\lambda_n = n.$$

Il n'y a que deux valeurs de λ_n pour lesquelles R_0 est satisfaite :

- $\lambda_n = 1$: ceci entraîne que $\lambda_1 = \lambda_2 = \dots = \lambda_{n-1} = 0$ et l'on obtient le n-uple solution : $(0, 0, 0, \dots, 0, 1)$.

- $\lambda_n = 0$: R_0 se réécrit :

$$R_1 : 1.\lambda_1 + 2\lambda_2 + \dots + (n-1)\lambda_{n-1} = n$$

Nous divisons alors les solutions restantes en deux groupes :

* celles pour lesquelles λ_1 est différent de 0.

* celles pour lesquelles λ_1 est nul.

a) $\lambda_1 \neq 0$ R_1 se réécrit alors :

$$R'_1 : 1.(\lambda_1 - 1) + 2.\lambda_2 + \dots + (n-1)\lambda_{n-1} = (n-1). \text{ Soit en posant } \lambda'_1 = \lambda_1 - 1.$$

$$1.\lambda'_1 + 2.\lambda_2 + \dots + (n-1)\lambda_{n-1} = (n-1) \text{ dont les solutions sont celles de } P_{n-1}.$$

Si $(a_1, a_2, \dots, a_{n-1})$ est une solution de P_{n-1} alors le n-uple

$$(a_1 + 1, a_2, \dots, a_{n-1}, 0) \text{ est solution de } P_n.$$

Un premier groupe de solutions de P_n est donc formé de tous les n-uples obtenus ainsi :

- dans chacun des $(n-1)$ uples solutions de P_{n-1} , on ajoute $1 = \lambda_1$ et on laisse les λ_i pour $i = 2, \dots, n-1$ inchangés.

- chaque $(n-1)$ -uple ainsi formé est transformé en n-uple en le complétant par la valeur de λ_n soit 0.

b) $\lambda_1 = 0$ R_1 devient :

$$R_2 : 2\lambda_2 + 3.\lambda_3 + \dots + (n-1)\lambda_{n-1} = n.$$

La seule valeur possible pour λ_{n-1} est alors 0 et R_2 s'écrit :

$$R_2 : 2\lambda_2 + 3.\lambda_3 + \dots + (n-2)\lambda_{n-2} = n.$$

Nous divisons les solutions de R2 en 2 groupes :

* celles pour lesquelles λ_2 est différent de 0.

* celles pour lesquelles λ_2 est nul.

a') $\lambda_2 \neq 0$ R2 s'écrit :

$$R_2' : 2(\lambda_2 - 1) + 3.\lambda_3 + \dots + (n-2)\lambda_{n-2} = n-2 \text{ soit avec } \lambda_2' = \lambda_2 - 1$$

$2\lambda_2' + 3\lambda_3 + \dots + (n-2)\lambda_{n-2} = n-2$ dont les solutions sont celles de P_{n-2} dans lesquelles λ_1 est nul.

Si $(0, a_2, a_3, \dots, a_{n-2})$ est une solution de P_{n-2} alors :

$(0, a_2 + 1, a_3, \dots, a_{n-2}, 0, 0)$ est une solution de P_n .

Un deuxième groupe de solutions de P_n est donc formé de tous les n-uples obtenus ainsi :

- on prend tous les (n-2)-uples solutions de P_{n-2} dans lesquels λ_1 est nul.
- dans chacun de ces (n-2)-uples on ajoute 1 à λ_2 et on laisse les λ_i $i = 3, \dots, n-2$ inchangés.
- chaque (n-2)-uple ainsi formé est transformé en n-uple en le complétant par les valeurs de λ_{n-1} et de λ_n soit 0,0.

b') $\lambda_2 = 0$ R2 devient :

$$R_3 : 3.\lambda_3 + 4.\lambda_4 + \dots + (n-2)\lambda_{n-2} = n$$

La seule valeur possible pour λ_{n-2} est alors 0 et R3 devient :

$$R_3 : 3.\lambda_3 + 4.\lambda_4 + \dots + (n-3)\lambda_{n-3} = n.$$

De nouveau on divise les solutions restantes en 2 groupes :

* celles pour lesquelles λ_3 est différent de 0.

* celles pour lesquelles λ_3 est nul.

etc...

L'algorithme se poursuit de cette façon jusqu'à ce que l'on ait utilisé dans la construction des solutions de P_n , les solutions du problème P_q où q est égal à :

- $n/2$ si n est pair

- $n\%2+1$ si n est impair.

En effet si n est pair, $n = 2k$, la relation de départ R_0 se transforme en R_1, R_2, \dots, R_k avec :

$$R_k : k \cdot \lambda_k = 2k$$

soit $k \cdot (\lambda_k - 1) = k$ dont les solutions sont celles du problème P_k dans lesquelles $\lambda_1, \lambda_2, \dots, \lambda_{k-1}$ sont nuls et auxquelles on rajoute 1 à λ_k .

Si $n = 2k+1$ R_0 se transforme en R_1, R_2, \dots, R_k avec :

$$R_k : k \cdot \lambda_k + (k+1) \lambda_{k+1} = 2k+1 \text{ d'où :}$$

a) Si $\lambda_k \neq 0$ $k(\lambda_k - 1) + (k+1)\lambda_{k+1} = k+1$ dont les solutions sont celles du problème P_{k+1} dans lesquelles $\lambda_1, \lambda_2, \dots, \lambda_{k-1}$ sont nuls et auxquelles on ajoute 1 à λ_{k-1} .

b) Si $\lambda_k = 0$ on a :

$(k+1)\lambda_{k+1} = 2k+1$ et il n'existe alors pas de valeurs de λ_{k+1} telles que ceci soit vérifié.

Cet algorithme nous permet donc de construire de façon récurrente les solutions d'un problème P_n pour n quelconque et c'est ce que nous avons fait pour n variant de 2 à 20.

V - PROCEDURE ALGOL

La procédure que nous donnons ici tient compte du fait que l'encombrement de la mémoire croît très vite avec l'ordre n du problème dont on recherche les solutions. De plus le mode de rangement des résultats que nous avons utilisé, est étroitement lié à l'organisation de la mémoire centrale du matériel dont nous disposons, I.B.M. 7044, en mots de 36 positions binaires directement adressables. Pour ces raisons nous nous sommes limités à traiter les problèmes d'ordre n inférieur ou égal à 30 et nous n'avons utilisé la procédure que pour n inférieur ou égal à 20.

A - RANGEMENT DES RESULTATS

1°) Pour un n -uple $(\lambda_1, \lambda_2, \dots, \lambda_n)$ solution du problème P_n , le nombre de λ_i non nuls est faible par rapport à n . Comme il n'est pas nécessaire de stocker les λ_i qui sont nuls, nous aurons en procédant ainsi un gain de place appréciable.

Pour savoir cependant quels sont les λ_i non nuls dans un n -uple solution de P_n , nous avons choisi de conserver une "trace" de ce n -uple de la façon suivante : à chaque n -uple $(\lambda_1, \lambda_2, \dots, \lambda_n)$ nous associons une mémoire d'un tableau TRACE. Nous divisons cette mémoire en 2 parties :

- une partie formée des 30 bits les plus à droite (numérotés de 1 à 30 de droite à gauche). Dans cette zone le bit n° k est égal à 1 si λ_k est différent de 0 dans le n -uple considéré et à 0 sinon.
- une partie formée des 6 bits restants, qui contiendra le nombre de bits non nuls de la partie droite de la mémoire, cette information étant utilisée lors de calculs d'adresses.

Exemple : pour $n = 5$ le 5-uple solution $(2,0,1,0,0)$ aura pour trace une mémoire dont la configuration sera :

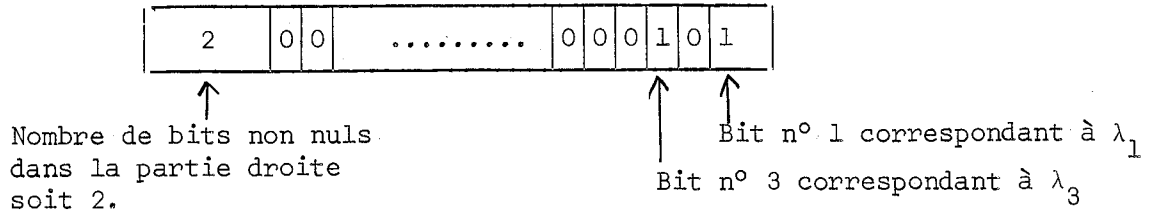


figure 1

2°) Ayant conservé la trace d'une solution il faut aussi stocker les valeurs des λ_i non nuls de cette solution. Ceci a été réalisé de la façon suivante : à chaque n-uple solution du problème P_n est associée une mémoire d'un tableau VALEUR. Cette mémoire est divisée en 7 zones de 5 bits numérotées de droite à gauche de 1 à 7, la zone n° p ayant pour valeur la valeur du λ_i correspondant au p^{ième} bit non nul de la trace du n-uple considéré (dans le cas où $n = 30$ le plus grand nombre de λ_i non nuls que l'on peut avoir dans une solution est de 7, d'ou le choix de 7 zones).

Exemple : Pour $n = 5$, au 5-uple solution $(2,0,1,0,0)$ correspondra une mémoire du tableau VALEUR dont la configuration sera :

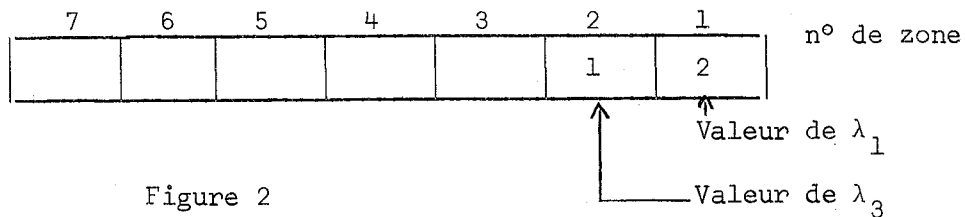


Figure 2

Nous voyons donc que les 6 bits de gauche de la mémoire représentée sur la figure 1 donne aussi le nombre de zones de 5 bits occupés dans la mémoire représentée sur la figure 2.

3°) La recherche des solutions du problème P_n exigeant la connaissance des solutions des problèmes précédents, il faut pouvoir accéder à ces solutions. Ceci se fait à l'aide d'un tableau ADRESSE, dont chaque mémoire est divisée en 3 zones numérotées 1,2,3 de droite à gauche.

La mémoire n° I du tableau ADRESSE ($2 \leq I \leq 30$) contiendra les informations suivantes :

- dans les 12 bits les plus à gauche sera rangé le nombre de solutions du problème n° I.
- dans les 12 bits suivant se trouvera l'adresse du tableau VALEUR à partir de laquelle sont rangées les valeurs numériques des λ_i apparaissant dans les i-uples solutions du problème I.
- dans les 12 bits des plus à droite, il y aura l'adresse du tableau TRACE à partir de laquelle sont rangées les traces des solutions du problème I.

Exemple : La mémoire n° 5 du tableau adresse aura, après le calcul des solutions du problème 5, la configuration :

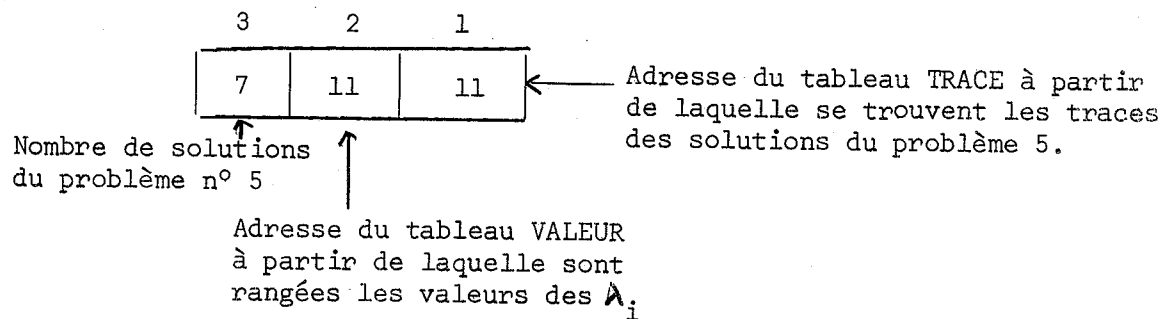


Figure 3

A titre d'exemple voici le contenu des premières mémoires de chacun des tableaux ADRESSE, TRACE et VALEURS après avoir calculé les solutions du problème 4.

Solutions du problème 2 : (0,1) (2,0)
 Solutions du problème 3 : (0,0,1) (1,1,0) (3,0,0)
 Solutions du problème 4 : (0,0,0,1) (1,0,1,0) (2,1,0,0)
 (0,2,0,0) (4,0,0,0)

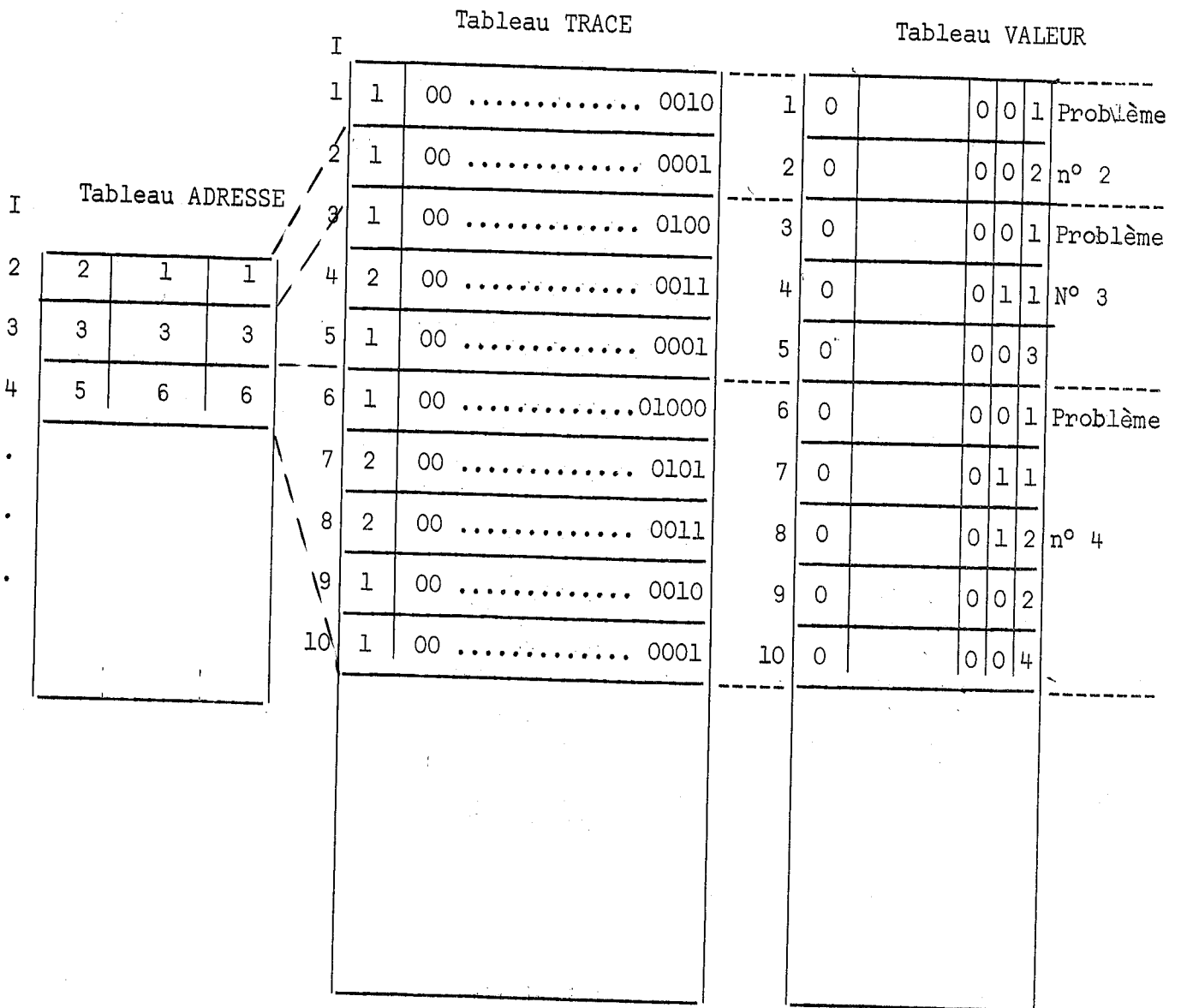


Figure 4

B - PROCEDURES UTILISEES

Cette méthode de rangement, bien que très intéressante du point de vue place utilisée, à toutefois l'inconvénient de nécessiter de nombreux calculs d'adresse. C'est pour cette raison que nous avons défini à l'intérieur du corps de procédure principal différentes procédures permettant de calculer des adresses du tableau VALEUR et du tableau TRACE.

a) entier procedure ADTRA (I,H) ; entier I,H ;

Cette procédure calcule l'adresse de la solution n° H du problème n° I dans le tableau TRACE et l'affecte à ADTRA.

b) entier procedure ADVAL(I,H) ; entier I,H ;

Cette procédure calcule l'adresse du tableau VALEUR à partir de laquelle les valeurs numériques de la solution H du problème I sont rangées.

c) procedure AJOUTZONE(NZ,DEPAR,MVA) ; entier NZ,DEPAR,MVA ;

Cette procédure recopie les NZ zones de 5 bits commençant à la mémoire DEPAR du tableau VALEUR dans la mémoire (ou la suite de mémoires) commençant à l'adresse MVA du même tableau VALEUR, après avoir décalé ces zones de 5 bits à gauche de 5 positions binaires et en mettant 1 dans la zone ainsi libérée.

Cette procédure nous sert lorsque, dans la construction des solutions du problème n, nous utilisons des solutions des problèmes n-1, n-2 etc... dans lesquelles nous voulons ajouter 1 à un λ_i qui y était nul. Le nombre de zones est donc augmenté et comme l'algorithme le montre, dans un tel cas les $\lambda_{i-1}, \lambda_{i-2}, \dots, \lambda_1$ sont nuls : pour respecter l'ordre d'apparition des λ_i non nuls de la solution dans le tableau TRACE et celui dans lequel apparaissent les valeurs de ces λ_i dans le tableau VALEUR, il faut bien décaler ces zones de 5 positions à gauche et mettre 1 dans la zone libérée.

d) procedure CØPIEZØNE (NZ,ADZ,MVA) ; entier NZ,ADZ,MVA ;

Cette procédure recopie les NZ zones de 5 bits commençant à l'adresse ADZ du tableau VALEUR, à partir de la mémoire MVA du tableau VALEUR. Cette procédure nous sert lorsque, dans la construction des solutions du problème n, nous utilisons des solutions des problèmes n-1, n-2, etc, dans lesquelles nous voulons ajouter 1 à un i qui n'était pas nul. Il n'y a pas de création de nouvelle zone dans le tableau VALEUR.

e) entier procedure ISOLE(NZ,ADR) ; entier NZ,ADR ;

Cette procédure nous donne le contenu de la zone n° NZ de la mémoire (ou de la suite de mémoires) débutant à la mémoire ADR du tableau VALEUR.

f) procedure CHARGE (MEM,P,X) ; entier MEM,P,X ;

Cette procédure ajoute X à la zone P de 12 bits (P = 1,2,3) de la mémoire MEM : nous l'utiliserons pour la mise à jour du contenu des mémoires du tableau ADRESSE ;

g) procedure CHARTRA(X,Y) ; entier X,Y ;

Cette procédure rajoute Y aux bits 31 à 36 de la mémoire X : elle sera utilisée pour la mise à jour du contenu de la partie gauche des mémoires du tableau TRACE ;

h) booleen procedure ZONUL(X,N1) ; entier X,N1 ;

Cette procédure prend la valeur vrai si les N1 bits de droite de la mémoire X sont nuls et faux dès que l'un d'eux est égal à 1.

Cette procédure nous sert lorsque nous recherchons les solutions du problème p dans lesquelles $\lambda_1, \lambda_2, \dots, \lambda_{n_1}$ sont nuls.

Dans toutes ces procédures, ainsi que dans le corps de procédure principal, nous avons utilisé les procédures booléennes élémentaires de l'équipe de logique de l'I.M.A.G. :

entier procedure ET(A,B) : intersection logique des mémoires A et B.
entier procedure OU(A,B) : union logique des mémoires A et B.
entier procedure TG(A,N) : translation gauche de la mémoire A de N positions binaires.
entier procedure TD(A,N) : translation droite de la mémoire A de N positions binaires.
entier procedure BIT(A,I) : vaut 0 si le bit N° I de la mémoire A est nul, 1 sinon.
entier procedure NG(A) : prend le complément de la mémoire A.

A l'aide de ces 6 procédures et de masques correspondant à la division des mémoires des tableaux ADRESSE, TRACE et VALEUR en plusieurs zones, il est possible à chaque instant d'isoler le contenu d'une de ces zones, sans détruire le contenu de la mémoire correspondante.

Enfin la tête de procédure principale est :

procedure LAMBDA SOL(ADRESSE,TRACE,VALEUR,N) ; entier tableau ADRESSE,TRACE,VALEUR ;
entier N ;

N indique ici l'ordre du problème que l'on veut résoudre, étant bien entendu que toutes les solutions des problèmes précédents seront d'abord construites et conservées comme nous l'avons indiqué. A cet effet les tableaux ADRESSE, TRACE et VALEUR sont initialisés avec les solutions du problème d'ordre 2 (cf. figure 4, page 16). La procédure construit alors les solutions du problème 3, puis 4 etc... jusqu'à N.

'DEBUT'

'COMMENTAIRE' CE PROGRAMME RECHERCHE, POUR UN ENTIER N FIXE, TOUTS
LES N-UPLES (L1, L2, ..., LN) OU LK EST UN ENTIER POSITIF
OU NUL, TELS QUE

$$1*L1+2*L2+\dots+N*LN = N.$$

POUR CELA ON CONSTRUIT LES SOLUTIONS DES PROBLEMES
3, 4, ..., N-1, N, LA RECHERCHE DES SOLUTIONS DU PROBLEME
D'ORDRE N NECESSITANT LA CONNAISSANCE DE CELLES DES
PROBLEMES D'ORDRE N-1, N-2, ..., P OU P EST EGAL A :

- N'/2 SI N EST PAIR
- N'/2 SI N EST IMPAIR.

NOUS UTILISERONS TROIS TABLEAUX :

- TABLEAU ADRESSE : CHAQUE MEMOIRE I EST DIVISEE EN 3
ZONES .
 - BITS 0 A 11 : NOMBRE DE SOLUTIONS DU
PROBLEME I .
 - BITS 12 A 23 : ADRESSE DANS LE TABLEAU TRACE A
PARTIR DE LAQUELLE SONT RANGEES LES TRACES
DES SOLUTIONS DU PROBLEME I .
 - BITS 24 A 35 : ADRESSE DANS LE TABLEAU VALEURS
A PARTIR DE LAQUELLE SONT RANGEES LES
VALEURS NUMERIQUES DES SOLUTIONS NON
NULLES DU PROBLEME I .
- TABLEAU TRACE : CHAQUE MEMOIRE EST DIVISEE EN 2
ZONES ET PORTE LA TRACE D'UN N-UPLE SOLUTION
DU PROBLEME A .
 - BITS 0 A 5 : NOMBRE DE BITS NON NULS DE LA
SECONDE ZONE
 - BITS 6 A 35 : LE BIT K EST EGAL A 1 SI LK EST
DIFFERENT DE 0 DANS LE N-UPLE SOLUTION,
0 AUTREMENT .
- TABLEAU VALEURS : CHAQUE MEMOIRE EST DIVISEE EN 7
ZONES DE 5 BITS, NUMEROTEES DE DROITE A GAUCHE,
AYANT POUR VALEURS LES VALEURS NUMERIQUES DES LK
NON NULS DE LA SOLUTION I DU PROBLEME N TRAITÉ.
LE NOMBRE DE ZONES AINSI OCCUPEES EST EGAL AU
CONTENU DES BITS 1 A 6 DE LA MEMOIRE DU
TABLEAU TRACE CORRESPONDANTE ::

'PROCEDURE' LAMBDA SOL (ADR, T, VAL, N) :: 'VALEUR' N ::

'ENTIER' N ::

'ENTIER' 'TABLEAU' ADR, T, VAL ::

'DEBUT' 'ENTIER' NPROB, MTR, MVA, H, NSOL, CASH1 A6, CASH7 A36, IP, IPMIN,
J, NZ, MP, ADZ, CT SOL, I, ADSOL, POINT ::

'ENTIER' 'TABLEAU' MASK. (1:3), MASQUE. (1:7), NUM,
MASKBIT. (1:30), ::

'ENTIER' 'PROCEDURE' ADTRA (I, H) :: 'VALEUR' I, H :: 'ENTIER' I, H ::

'COMMENTAIRE' CETTE PROCEDURE DONNE L'ADRESSE DE LA SOLUTION H DU
PROBLEME I DANS LE TABLEAU T ::

ADTRA := ET (MASK. (1), ADR. (I)) + H - 1 ::

'ENTIER' 'PROCEDURE' ADVAL (I, H) :: 'VALEUR' I, H :: 'ENTIER' I, H ::

'COMMENTAIRE' CETTE PROCEDURE DONNE L'ADRESSE DU TABLEAU VAL A
PARTIR DE LAQUELLE LES VALEURS NUMERIQUES DE LA SOLUTION H DU
PROBLEME I SONT RANGEES ::

'DEBUT' 'ENTIER' A1, A2, A3, A4, A5 ::

A2 := TD (ET (ADR. (I), MASK. (2)), 12) ::

6131,0293,0035 GUYOT J.L

EDITION ALGOL LANDA

```

'SI' H = 1 'ALORS' ADVAL := A2 'SINON'
'DEBUT'
A1 := TD(ET(T.(ADTRA(I,1)),CASH1A6),30) ::
A3 := A1/'7' ::
'SI' A3*7 = A1 'ALORS' A2 := A2+A3-1 'SINON' A2 := A2+A3 ::
'POUR' A1 := 2 'PAS' 1 'JUSQUA' H-1 'FAIRE'
'DEBUT' A3:=ADTRA(I,A1) ::
A4:=TD(ET(T.(A3),CASH1A6),30) ::
A5:=A4/'7' ::
'SI' A5*7 = A4 'ALORS' A2 := A2+A5 'SINON' A2 := A2+A5+1 ::
'FIN' ::
ADVAL := A2 + 1 ::
'FIN' ::
'FIN' ADVAL ::
'PROCEDURE' CT(X) :: 'ENTIER' X ::
X:=X+1 ::
'PROCEDURE' AJOUTZONE(NZ,DEPAR,MVA) :: 'ENTIER' NZ,DEPAR,MVA ::
'COMMENTAIRE' CETTE PROCEDURE RECOPIE LES NZ ZONES DE 5 BITS
COMMENCANT A LA MEMOIRE DEPAR DU TABLEAU VAL A PARTIR DE LA
MEMOIRE MVA DU TABLEAU VAL, EN DECALANT CES ZONES DE 5 BITS
A GAUCHE ET EN METTANT 1 DANS LA ZONE LIBEREE. SI LE NOMBRE DE
ZONES EST UN MULTIPLE DE 7, ON FAIT PROGRESSER MVA ::
'DEBUT' 'ENTIER' D,E,F,G,I ::
D:=NZ/'7' :: 'SI' D*7 'NONEG' NZ 'ALORS' D:=D+1 ::
F:=MASKBIT.(1). ::
'POUR' I:=1 'PAS' 1 'A' D 'FAIRE'
'DEBUT'
G:=VAL.(DEPAR+I-1). ::
E:=TG(ET(G,NG(MASQUE.(7).)),5) ::
VAL.(MVA+I-1).:=DU(E,F) ::
F:=TD(ET(MASQUE.(7),G),30) ::
'FIN' ::
MVA:=MVA+D ::
'SI' ET(F,MASQUE.(1).) 'NONEG' 0 'ALORS'
'DEBUT' VAL.(MVA).:=F ::
CT(MVA) ::
'FIN' ::
'FIN' ::
'PROCEDURE' COPIEZONE(NZ,ADZ,MVA) ::
'ENTIER' NZ,ADZ,MVA ::
'COMMENTAIRE' CETTE PROCEDURE RECOPIE LES NZ ZONES DE 5 BITS
COMMENCANT A LA MEMOIRE ADZ DU TABLEAU VAL, A PARTIR
DE LA MEMOIRE MVA DU TABLEAU VAL. MVA PROGRESSE EN FONCTION
DU NOMBRE DE MEMOIRES AINSI REMPLIES ::
'DEBUT' 'ENTIER' K1,K2 ::
K1:=NZ/'7' ::
'SI' K1*7 'NONEG' NZ 'ALORS' K1:=K1+1 ::
VAL.(MVA).:=VAL.(ADZ).+1 :: CT(MVA) ::
'POUR' K2:=2 'PAS' 1 'JUSQUA' K1 'FAIRE'
'DEBUT'
VAL.(MVA).:=VAL.(ADZ+K2-1). ::
CT(MVA) ::
'FIN' ::
'FIN' ::
'ENTIER' 'PROCEDURE' ISOLE(NZ,ADRESSE) :: 'ENTIER' NZ,ADRESSE ::

```


6131, 0293, 0035 GUYOT J.L

EDITION ALGOL LANDA

```

*COMMENTAIRE* CETTE PROCEDURE DONNE LE CONTENU DE LA ZONE
  N0 NZ DE LA ZONE DE MEMOIRES DEBUTANT A VAL(ADRESSE),
  CES MEMOIRES ETANT DIVISEES EN 7 ZONES DE 5 BITS
  NUMEROTEES DE DROITE A GAUCHE ET DE HAUT EN BAS ::
*DEBUT* 'ENTIER' C1,C2,C3,C4 ::
  C1:=NZ*/7 :: C4:=NZ-C1*7 ::
  C2:=VAL.(ADRESSE+C1). ::
  C3:=TD(ET(C2,MASQUE.(C4).),(C4-1)*5) ::
  ISOLE:=C3 ::
*FIN* ::
*PROCEDURE* CHARGE(MEM,P,X) :: 'ENTIER' MEM,P,X ::
*COMMENTAIRE* CETTE PROCEDURE AJOUTE X A LA ZONE P (P=1,2,3)
  DE 12 BITS DE LA MEMOIRE MEM ::
*DEBUT* 'ENTIER' D1,D2,D3 ::
  D3:=(P-1)*12 ::
  D1:=ET(MEM,MASK.(P).) ::
  D2:=ET(MEM,NG(MASK.(P).)) ::
  D1:=TG( TD(D1,D3)+X,D3) ::
  MEM:=OU(D1,D2) ::
*FIN* ::
*PROCEDURE* CHARTRA(X,Y) :: 'ENTIER' X,Y ::
*COMMENTAIRE* CETTE PROCEDURE RAJOUTE Y AUX BITS 1 A 6 DE LA
  MEMOIRE X ::
*DEBUT* 'ENTIER' E1,E2 ::
  E1:=TD(ET(X,CASHIA6),30) ::
  E1:=TG(E1+Y,30) ::
  X:=OU(E1,ET(X,CASH7A36)) ::
*FIN* ::
*BOOLEEN* 'PROCEDURE' ZONLL(X,N1) :: 'ENTIER' X,N1 ::
*COMMENTAIRE* CETTE PROCEDURE PREND LA VALEUR VRAI SI LES
  N1 BITS DE DROITE DE LA MEMOIRE X SONT NULS, ET FAUX DES
  QUE L'UN D'EUX EST EGAL A 1 ::
*DEBUT* 'ENTIER' I :: 'BOOLEEN' BC :: BC:='VRAI' :: I:=0 ::
  'POUR' I:=I+1 'TANT QUE' I<N1 'ET' BC 'FAIRE'
  BO:=BO 'ET' (BIT(X,I)=0) ::
  ZONLL:=BO ::
*FIN* ::
DEFINIMASK: MASK.(1).:=4095 :: MASK.(2).:=TG(4095,12) ::
MASK.(3).:=TG(4095,24) ::
CASHIA6:=TG(63,30) :: CASH7A36:=NG(CASHIA6) ::
MASQUE.(1).:=31 ::
'POUR' I:=2 'PAS' 1 'JUSQUA' 7 'FAIRE'
  MASQUE.(I).:=TG(31,(I-1)*5) ::
MASKBIT.(1).:=1 ::
'POUR' I:=2 'PAS' 1 'JUSQUA' 30 'FAIRE'
  MASKBIT.(I).:=TG(1,I-1) ::
'POUR' I:=1 'PAS' 1 'JUSQUA' N 'FAIRE'
  NUM.(I).:=I ::
INITIALISATIONS:
CHARGE(ADR.(2).,1,1) :: CHARGE(ADR.(2).,2,1) ::
CHARGE(ADR.(2).,3,2) ::
CHARTRA(T.(1).,1) :: T.(1).:=OU(T.(1).,MASKBIT.(2).) ::
  VAL.(1).:=1 ::
CHARTRA(T.(2).,1) :: T.(2).:=OU(T.(2).,MASKBIT.(1).) ::
  VAL.(2).:=2 ::

```

6131,0293,0035 GUYOT J.L

EDITION ALGOL LANCA

```

MTR:=MVA:=3 ::
'POUR' NPROB:=3 'PAS' 1 'JUSQUA' N 'FAIRE'
'DEBUT' NSOL:=0 ::
CHARGE(ADR.(NPROB),,1,MTR) ::
CHARGE(ADR.(NPROB),,2,MVA) ::
CHARTRA(T.(MTR),,1) ::
T.(MTR).:=OL(T.(MTR),MASKBIT.(NPROB).) ::
VAL.(MVA).:=1 :: CT(MTR) :: CT(MVA) :: CT(NSOL) ::
H:=NPROB/'2 :: 'SI' H*2=NPROB 'ALCRS' IPMIN:=H
'SINON' IPMIN:=H+1 ::
IP:=NPROB ::
'POUR' IP:=IP-1 'ANTIQUE' IP 'SUPEG' IPMIN 'FAIRE'
'DEBUT'
CTSOL:=TD(ET(ADR.(IP),,MASK.(3).),24) ::
'SI' 8000-MTR 'INFEG' 2*CTSOL 'ALCRS' 'ALLERA' RESULTATS ::
'SI' IP=NPROB-1 'ALCRS'
'DEBUT' 'POUR' J:=1 'PAS' 1 'JUSQUA' CTSOL 'FAIRE'
'DEBUT' MP:=1.(ADTRA(IP,J)). ::
NZ:=TD(ET(CASHIA6,MP),30) ::
ADZ:=ADVAL(IP,J) ::
'SI' ET(MP,MASKBIT.(1).)=0 'ALCRS'
'DEBUT' CHARTRA(MP,1) ::
MP:=OL(MP,MASKBIT.(1).) ::
T.(MTR).:=MP :: CT(MTR) ::
AJOLTZONE(NZ,ADZ,MVA) ::
'FIN' 'SINON'
'DEBUT' T.(MTR).:=MP ::
COPIEZONE(NZ,ADZ,MVA) ::
CT(MTR) ::
'FIN' ::
'FIN' ::
NSOL:=NSOL+CTSOL :: 'ALLERA' FINPBMCINSI ::
'FIN' ::
'POUR' J:=1 'PAS' 1 'JUSQUA' CTSOL 'FAIRE'
'DEBUT' ADSOL:=ADTRA(IP,J) ::
'SI' ZONUL(T.(ADSOL),,NPROB-IP-1) 'ALCRS'
'DEBUT' MP:=1.(ADSOL). ::
NZ:=TD(ET(CASHIA6,MP),30) ::
ADZ:=ADVAL(IP,J) ::
'SI' BIT(MP,NPROB-IP)=0 'ALCRS'
'DEBUT'
T.(MTR).:=OL(MP,MASKBIT.(NPROB-IP).) ::
CHARTRA(T.(MTR),,1) :: CT(MTR) ::
AJOLTZONE(NZ,ADZ,MVA) ::
'FIN' 'SINON'
'DEBUT' T.(MTR).:=MP ::
COPIEZONE(NZ,ADZ,MVA) ::
CT(MTR) ::
'FIN' ::
NSOL:=NSOL+1 ::
'FIN' SI ::
'FIN' BOUCLE J :: FINPBMCINSI :
'FIN' BOUCLE IP :: CHARGE(ADR.(NPROB),,3,NSOL) ::
'FIN' BOUCLE NPROB ::
RESULTATS:

```

6131, 0293, 0035 GUYDT J.L.

EDITION ALGOL LANCA

```

SP: MODELE(''(1H1)'')::
SORTIE(6,SP) :: POINT := 0 ::
'POUR' I := 2 'PAS' 1 'A' N 'FAIRE'
'DEBUT' 'ENTIER' 'TABLEAU' TAB.(1:I). :: 'ENTIER' BNUL ::
NSOL := TD(ET(ADR.(I)., MASK.(3).), 24) ::
'SI' POINT + 4 'SUPEG' 52 'ALORS'
'DEBUT' SORTIE(6,SP) :: POINT := 0 :: 'FIN' ::
SORTIE(66,M3,1,I,1,NSOL) ::
POINT := POINT + 4 ::
'SI' POINT 'SUPEG' 58 'ALORS'
'DEBUT' SORTIE(6,SP) :: POINT:=0 :: 'FIN' ::
'POUR' J:=1 'PAS' 1 'JUSQUA' NSOL 'FAIRE'
'DEBUT' MP:=ADTRA(I,J) :: ADSOL:=ADVAL(I,J) ::
NZ:=TD(ET(T.(MP)., CASH1A6), 30) ::
BNUL:=C ::
'POUR' H:=1 'PAS' 1 'JUSQUA' I 'FAIRE'
'SI' BIT(T.(MP).,H)=0 'ALORS' TAB.(H). := 0
'SINON'
'DEBUT' BNUL:=BNUL+1 ::
TAB.(H).:=ISOLE(BNUL,ADSOL) ::
'FIN' ::
SORTIE(66,M2,I,TAB.(1).) ::
POINT := POINT + 1 ::
'SI' POINT 'SUPEG' 58 'ALORS'
'DEBUT' SORTIE(6,SP) :: POINT:=0 :: 'FIN' ::
'FIN' ::
'FIN' RESULTATS ::
SORTIE(6,SP) ::
M2: MODELE(''(1H,30I4)'') ::
M3: MODELE(''(1HC,13H PROBLEME NO ,I4,23H NOMBRE DE SOLUTIONS ,I
4//)'') ::
'FIN' LAMBDA SOL ::
'ENTIER' N :: N:= 15 ::
'DEBUT' 'ENTIER' 'TABLEAU' ADRESSE.(2:N+1)., TRACE.(1:8000).,
VALEURS.(1:8000). ::
LAMBDA SOL (ADRESSE, TRACE, VALEURS, N) ::
'FIN' ::
'FIN' DU PROGRAMME ::
FALGOL

```

VI - RESULTATS

Avant de donner les résultats obtenus, nous préciserons que nous avons utilisé cette procédure pour $N = 20$ et que le calcul effectif des solutions du problème 3 à celles du problème 20 nécessite environ 8 minutes de calcul. En ce qui concerne la place utilisée pour conserver les résultats nous ferons remarquer que le problème 20 à plus de 600 20-uples solutions ce qui occuperait 12.000 mémoires si nous avions rangé ces solutions à raison de un λ_i par mémoire, alors que par notre méthode nous n'en utilisons que 1200.

Nous ne donnerons ici que les formules obtenues pour $n \leq 8$ car nous pensons que si l'on veut effectivement calculer les coefficients d'un polynôme à l'aide des formules (1.3), il est préférable de connaître les valeurs des λ_i intervenant dans ces formules plutôt que d'en avoir l'expression littérale complète. Il nous semble donc plus intéressant de donner la table des i -uples $(\lambda_1, \lambda_2, \dots, \lambda_i)$ solutions de $1.\lambda_1 + 2.\lambda_2 + \dots + i.\lambda_i = i$ pour les valeurs de i comprises entre 2 et 15.

a) Formules littérales

Nous donnons ici les expressions des p_i/p_0 .

$$p_1/p_0 = -s_1$$

$$p_2/p_0 = \frac{1}{2} s_1^2 - \frac{1}{2} s_2$$

$$p_3/p_0 = \frac{1}{2} s_1 s_2 - \left(\frac{1}{3!} s_1^3 + \frac{1}{3} s_3 \right)$$

$$p_4/p_0 = \frac{1}{4!} s_1^4 + \frac{1}{1 \cdot 3} s_1 s_3 + \frac{1}{2 \cdot 2!} s_2^2 - \left(\frac{s_1^2 s_2}{2 \cdot 2!} + \frac{1}{4} s_4 \right)$$

$$p_5/p_0 = \frac{1}{4} s_1 s_4 + \frac{1}{2 \cdot 3} s_2 s_3 + \frac{1}{2 \cdot 3!} s_1^3 s_2 - \left(\frac{1}{5!} s_1^5 + \frac{1}{3 \cdot 2!} s_1^2 s_3 + \frac{1}{5} s_5 \right)$$

$$p_6/p_0 = \frac{s_1^6}{6!} + \frac{1}{3 \cdot 3!} s_1^3 s_3 + \frac{1}{5} s_1 s_5 + \frac{1}{2^2 (2!)^2} s_1^2 s_2^2 + \frac{1}{3 \cdot 2!} s_3^2 + \frac{1}{2 \cdot 4} s_2 s_4$$

$$- \left(\frac{1}{2 \cdot 4!} s_1^4 s_2 + \frac{1}{4 \cdot 2!} s_1^2 s_4 + \frac{1}{2 \cdot 3!} s_2^3 + \frac{1}{2 \cdot 3} s_1 s_2 s_3 + \frac{1}{6} s_6 \right)$$

$$p_7/p_0 = \frac{1}{6} s_1 s_6 + \frac{s_1 s_3^2}{3! 2^3} + \frac{s_1^2 s_2 s_3}{2! 2 \cdot 3} + \frac{s_1^3 s_4}{3! 4} + \frac{s_1^5 s_2}{3! 4} + \frac{s_1 s_2}{5! 2} + \frac{s_2 s_5}{2 \cdot 5} + \frac{s_3 \cdot s_4}{3 \cdot 4}$$

$$- \left(\frac{s_1 s_2 s_4}{2 \cdot 4} + \frac{s_1 s_3^2}{3^2 2!} + \frac{s_1^2 s_5}{2! 5} + \frac{s_1^3 s_2^2}{3! 2! 2^2} + \frac{s_1^4 s_3}{4! 3} + \frac{s_1^7}{7!} + \frac{s_2^2 s_3}{2! 3! 2^2} + \frac{s_7}{7} \right)$$

$$p_8/p_0 = \frac{s_1^8}{8!} + \frac{s_1^5 s_3}{5! 3} + \frac{s_1^4 s_2^2}{4! 2! 2^2} + \frac{s_1^3 s_5}{3! 5} + \frac{s_1^2 s_2 s_4}{2! 2 \cdot 4} + \frac{s_1^2 s_3^2}{3^2 2! 2!} + \frac{s_1 s_2^2 s_3}{2! 2^2 3} + \frac{s_1 s_7}{7} + \frac{s_2 s_6}{2 \cdot 6} +$$

$$+ \frac{s_3 s_5}{3 \cdot 5} + \frac{s_4^2}{4 \cdot 2!} + \frac{s_4^4}{2^4 \cdot 4!}$$

$$- \left(\frac{s_8}{8} + \frac{s_1^6 s_2}{6! 2} + \frac{s_1^4 s_4}{4! 4} + \frac{s_1^3 s_2 s_3}{3! 2 \cdot 3} + \frac{s_1^2 s_6}{2! 6} + \frac{s_1^2 s_3^3}{2^3 2! 3!} + \frac{s_2^2 s_4}{4 \cdot 2^2 \cdot 2!} + \frac{s_2 s_3^2}{2 \cdot 3^2 \cdot 2!} \right)$$

$$+ \frac{s_1 s_2 s_5}{2 \cdot 5} + \frac{s_1 s_3 s_4}{3 \cdot 4}$$

b) Tableau des i -uples solutions de $1 \cdot \lambda_1 + 2 \cdot \lambda_2 + \dots + i \cdot \lambda_i = i$ pour $i = 2$.

PROBLEME NO 2 NOMBRE DE SOLUTIONS 2

0	1
2	0

PROBLEME NO 3 NOMBRE DE SOLUTIONS 3

0	0	1
1	1	0
3	0	0

PROBLEME NO 4 NOMBRE DE SOLUTIONS 5

0	0	0	1
1	0	1	0
2	1	0	0
4	0	0	0
0	2	0	0

PROBLEME NO 5 NOMBRE DE SOLUTIONS 7

0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	1	0	0	0
5	0	0	0	0
1	2	0	0	0
0	1	1	0	0

PROBLEME NO 6 NOMBRE DE SOLUTIONS 11

0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	1	0	0	0
4	1	0	0	0	0
6	0	0	0	0	0
2	2	0	0	0	0
1	1	1	0	0	0
0	1	0	1	0	0
0	3	0	0	0	0
0	0	2	0	0	0

PROBLEME NO 7 NOMBRE DE SOLUTIONS 15

0	0	0	0	C	C	1
1	0	0	0	0	1	0
2	0	0	0	1	C	C
3	0	0	1	C	C	C
4	0	1	0	C	C	C
5	1	0	0	C	C	C
7	0	0	0	C	C	C
3	2	0	0	C	C	C
2	1	1	0	C	C	C
1	1	0	1	C	C	C
1	3	0	0	C	C	C
1	0	2	0	C	C	C
0	1	0	0	1	C	C
0	2	1	0	0	0	0
0	0	1	1	C	C	C

PROBLEME NO 8 NOMBRE DE SOLUTIONS 22

0	0	0	0	C	C	C	1
1	0	0	0	0	0	1	C
2	0	0	0	C	1	C	C
3	0	0	0	1	0	C	C
4	0	0	1	C	C	C	C
5	0	1	0	0	0	C	C
6	1	0	0	C	C	C	C
8	0	0	0	C	0	C	C
4	2	0	0	C	C	C	C
3	1	1	0	0	0	C	C
2	1	0	1	C	C	C	C
2	3	0	0	C	0	C	C
2	0	2	0	C	C	C	C
1	1	0	0	1	0	C	C
1	2	1	0	C	C	C	C
1	0	1	1	C	C	C	C
0	1	0	0	C	1	C	C
0	2	0	1	0	0	C	C
0	4	0	0	C	C	C	C
0	1	2	0	C	C	0	C
0	0	1	0	1	C	C	C
0	0	0	2	C	0	C	C

PROBLEME NO 9 NOMBRE DE SOLUTIONS 30

0	0	0	0	C	C	C	C	1
1	0	0	0	0	0	0	1	0
2	0	0	0	C	C	1	C	0
3	0	0	0	C	1	0	C	0
4	0	0	0	1	C	C	C	0
5	0	0	1	0	C	0	0	0
6	0	1	0	C	C	C	C	0
7	1	0	0	0	C	C	C	0
9	0	0	0	C	C	C	C	0

5	2	0	0	0	0	C	C	0
4	1	1	0	0	0	C	C	0
3	1	0	1	C	C	0	0	0
3	3	0	0	C	C	C	0	0
3	0	2	0	C	C	C	C	0
2	1	0	0	1	0	0	0	0
2	2	1	0	0	C	C	C	0
2	0	1	1	C	C	C	C	0
1	1	0	0	C	1	C	C	0
1	2	0	1	C	0	0	C	0
1	4	0	0	C	0	C	0	0
1	1	2	0	C	0	C	C	0
1	0	1	0	1	C	C	C	0
1	0	0	2	0	C	C	C	0
0	1	0	0	0	0	1	0	0
0	2	0	0	1	0	C	0	0
0	3	1	0	C	C	C	C	0
0	1	1	1	C	C	C	C	0
0	0	1	0	0	1	0	0	0
0	0	3	0	C	0	C	C	0
0	0	0	1	1	0	C	0	0

PROBLEME NO 10 NOMBRE DE SOLUTIONS 42

0	0	0	0	0	0	C	0	0	1
1	0	0	0	C	0	C	C	1	0
2	0	0	0	0	0	C	1	0	0
3	0	0	0	C	C	1	C	0	0
4	0	0	0	0	1	0	0	0	0
5	0	0	0	1	0	C	C	0	0
6	0	0	1	0	0	0	0	0	0
7	0	1	0	C	C	C	C	0	0
8	1	0	0	0	0	C	0	0	0
10	0	0	0	C	C	C	C	0	0
6	2	0	0	0	0	0	0	0	0
5	1	1	0	C	C	C	C	0	0
4	1	0	1	0	0	0	C	0	0
4	3	0	0	C	C	C	C	0	0
4	0	2	0	0	C	C	C	0	0
3	1	0	0	1	C	C	0	0	0
3	2	1	0	0	0	0	C	0	0
3	0	1	1	C	0	C	C	0	0
2	1	0	0	C	1	C	0	0	0
2	2	0	1	C	C	C	C	0	0
2	4	0	0	0	0	0	0	0	0
2	1	2	0	0	C	C	0	0	0
2	0	1	0	1	C	0	C	0	0
2	0	0	2	C	0	0	C	0	0
1	1	0	0	0	C	1	C	0	0
1	2	0	0	1	C	0	0	0	0
1	3	1	0	C	0	0	C	0	0
1	1	1	1	C	C	C	C	0	0
1	0	1	0	0	0	0	0	0	0
1	0	3	0	0	0	0	0	0	0
1	0	0	1	1	C	C	0	0	0
0	1	0	0	C	C	C	1	0	0
0	2	0	0	0	1	0	0	0	0

0	3	0	1	0	0	0	0	0	0
0	5	0	0	0	0	0	0	0	0
0	2	2	0	0	0	0	0	0	0
0	1	1	0	1	0	0	0	0	0
0	1	0	2	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0
0	0	2	1	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0
0	0	0	0	2	0	0	0	0	0

PROBLEME NO 11 NOMBRE DE SOLUTIONS 56

0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	1	0	0
3	0	0	0	0	0	0	1	0	0	0
4	0	0	0	0	0	1	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	1	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0	0	0
8	0	1	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0
7	2	0	0	0	0	0	0	0	0	0
6	1	1	0	0	0	0	0	0	0	0
5	1	0	1	0	0	0	0	0	0	0
5	3	0	0	0	0	0	0	0	0	0
5	0	2	0	0	0	0	0	0	0	0
4	1	0	0	1	0	0	0	0	0	0
4	2	1	0	0	0	0	0	0	0	0
4	0	1	1	0	0	0	0	0	0	0
3	1	0	0	0	1	0	0	0	0	0
3	2	0	1	0	0	0	0	0	0	0
3	4	0	0	0	0	0	0	0	0	0
3	1	2	0	0	0	0	0	0	0	0
3	0	1	0	1	0	0	0	0	0	0
3	0	0	2	0	0	0	0	0	0	0
2	1	0	0	0	0	1	0	0	0	0
2	2	0	0	1	0	0	0	0	0	0
2	3	1	0	0	0	0	0	0	0	0
2	1	1	1	0	0	0	0	0	0	0
2	0	1	0	0	1	0	0	0	0	0
2	0	3	0	0	0	0	0	0	0	0
2	0	0	1	0	1	0	0	0	0	0
1	1	0	0	0	0	0	1	0	0	0
1	2	0	0	0	1	0	0	0	0	0
1	3	0	1	0	0	0	0	0	0	0
1	5	0	0	0	0	0	0	0	0	0
1	2	2	0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	0	0	0	0
1	1	0	2	0	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0
1	0	2	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0
1	0	0	0	2	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0
0	2	0	0	0	0	1	0	0	0	0

0	3	0	0	1	C	C	C	C	0	0	0
0	4	1	0	0	C	C	C	C	0	0	0
0	2	1	1	0	C	C	C	C	0	0	0
0	1	1	0	0	C	1	C	C	0	0	0
0	1	3	0	0	C	C	C	C	0	0	0
0	1	0	1	1	C	C	C	C	0	0	0
0	0	1	0	0	C	C	C	1	0	0	0
0	0	2	0	0	1	C	C	C	0	0	0
0	0	1	2	0	C	C	C	C	0	0	0
0	0	0	1	1	C	C	1	C	0	0	0
0	0	0	0	1	1	1	C	C	0	0	0

PROBLEME NO 12 NOMBRE DE SOLUTIONS 77

0	0	0	0	C	C	C	C	0	0	0	1
1	0	0	0	C	C	C	C	0	0	1	0
2	0	0	0	C	C	C	C	0	1	0	0
3	0	0	0	C	C	C	C	1	0	0	0
4	0	0	0	C	C	C	1	C	0	0	0
5	0	0	0	C	C	1	C	C	0	0	0
6	0	0	0	C	1	C	C	C	0	0	0
7	0	0	0	1	C	C	C	C	0	0	0
8	0	0	1	C	C	C	C	C	0	0	0
9	0	1	0	C	C	C	C	C	0	0	0
10	1	0	0	C	C	C	C	C	0	0	0
12	0	0	0	C	C	C	C	C	0	0	0
8	2	0	0	C	C	C	C	C	0	0	0
7	1	1	0	C	C	C	C	C	0	0	0
6	1	0	1	C	C	C	C	C	0	0	0
6	3	0	0	C	C	C	C	C	0	0	0
6	0	2	0	C	C	C	C	C	0	0	0
5	1	0	0	1	C	C	C	C	0	0	0
5	2	1	0	C	C	C	C	C	0	0	0
5	0	1	1	C	C	C	C	C	0	0	0
4	1	0	0	C	1	C	C	C	0	0	0
4	2	0	1	C	C	C	C	C	0	0	0
4	4	0	0	C	C	C	C	C	0	0	0
4	1	2	0	C	C	C	C	C	0	0	0
4	0	1	0	1	C	C	C	C	0	0	0
4	0	0	2	C	C	C	C	C	0	0	0
3	1	0	0	C	C	1	C	C	0	0	0
3	2	0	0	1	C	C	C	C	0	0	0
3	3	1	0	C	C	C	C	C	0	0	0
3	1	1	1	C	C	C	C	C	0	0	0
3	0	1	0	C	1	C	C	C	0	0	0
3	0	3	0	C	C	C	C	C	0	0	0
3	0	0	1	1	C	C	C	C	0	0	0
2	1	0	0	C	C	C	1	C	0	0	0
2	2	0	0	C	1	C	C	C	0	0	0
2	3	0	1	C	C	C	C	C	0	0	0
2	5	0	0	C	C	C	C	C	0	0	0
2	2	2	0	C	C	C	C	C	0	0	0
2	1	1	0	1	C	C	C	C	0	0	0
2	1	0	2	C	C	C	C	C	0	0	0
2	0	1	0	C	C	1	C	C	0	0	0
2	0	2	1	C	C	C	C	C	0	0	0
2	0	0	1	C	1	C	C	C	0	0	0

2	0	0	0	2	C	C	C	0	0	0	0
1	1	0	0	C	C	C	C	1	0	0	0
1	2	0	0	C	C	1	C	0	0	0	0
1	3	0	0	1	C	C	C	0	0	0	0
1	4	1	0	C	C	C	C	0	0	0	0
1	2	1	1	C	C	C	C	0	0	0	0
1	1	1	0	C	1	C	C	0	0	0	0
1	1	3	0	C	C	C	C	0	0	0	0
1	1	0	1	1	C	C	C	0	0	0	0
1	0	1	0	C	C	C	1	0	0	0	0
1	0	2	0	1	0	0	0	0	0	0	0
1	0	1	2	C	C	C	0	0	0	0	0
1	0	0	1	C	C	1	C	0	0	0	0
1	0	0	0	1	1	C	C	0	0	0	0
0	1	0	0	0	0	0	0	0	1	0	0
0	2	0	0	0	0	0	1	0	0	0	0
0	3	0	0	C	1	C	C	0	0	0	0
0	4	0	1	C	C	C	C	0	0	0	0
0	6	0	0	0	C	C	C	0	0	0	0
0	3	2	0	0	0	0	0	0	0	0	0
0	2	1	0	1	C	C	C	0	0	0	0
0	2	0	2	C	C	C	C	0	0	0	0
0	1	1	0	0	C	1	C	0	0	0	0
0	1	2	1	C	0	0	C	0	0	0	0
0	1	0	1	C	1	C	C	0	0	0	0
0	1	0	0	2	C	C	C	0	0	0	0
0	0	1	0	0	C	C	C	1	0	0	0
0	0	2	0	0	1	0	0	0	0	0	0
0	0	4	0	C	C	0	C	0	0	0	0
0	0	1	1	1	C	C	C	0	0	0	0
0	0	0	1	C	C	C	1	0	0	0	0
0	0	0	3	0	0	0	0	0	0	0	0
0	0	0	0	1	C	1	C	0	0	0	0
0	0	0	0	C	2	0	C	0	0	0	0

PROBLEME NO 13 NOMBRE DE SOLUTIONS 101

0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	C	C	C	C	0	0	0	1	0
2	0	0	0	0	0	0	0	0	0	1	0	0
3	0	0	0	C	C	C	C	0	1	0	0	0
4	0	0	0	0	0	C	0	1	0	0	0	0
5	0	0	0	0	C	C	1	C	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0
7	0	0	0	C	1	C	C	0	0	0	0	0
8	0	0	0	1	0	0	0	0	0	0	0	0
9	0	0	1	0	C	C	C	0	0	0	0	0
10	0	1	0	0	C	C	C	0	0	0	0	0
11	1	0	0	C	C	C	C	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0
9	2	0	0	C	C	C	C	0	0	0	0	0
8	1	1	0	C	C	0	0	0	0	0	0	0
7	1	0	1	C	C	C	C	0	0	0	0	0
7	3	0	0	0	0	C	0	0	0	0	0	0
7	0	2	0	C	C	C	0	0	0	0	0	0
6	1	0	0	1	C	C	C	0	0	0	0	0
6	2	1	0	C	C	C	C	0	0	0	0	0

0	1	0	0	C	C	C	C	C	0	1	0	0
0	2	0	0	C	C	C	C	1	0	0	0	0
0	3	0	0	C	C	1	C	0	0	0	0	0
0	4	0	0	1	C	0	C	0	0	0	0	0
0	5	1	0	C	C	C	C	0	0	0	0	0
0	3	1	1	C	C	C	C	0	0	0	0	0
0	2	1	0	C	1	C	C	0	0	0	0	0
0	2	3	0	C	C	0	0	0	0	0	0	0
0	2	0	1	1	C	C	C	0	0	0	0	0
0	1	1	0	C	C	C	1	C	0	0	0	0
0	1	2	0	1	C	C	C	0	0	0	0	0
0	1	1	2	C	0	0	0	0	0	0	0	0
0	1	0	1	C	C	1	C	C	0	0	0	0
0	0	1	0	1	C	C	C	0	0	0	0	0
0	0	2	0	C	C	1	C	0	0	0	0	0
0	0	3	1	C	C	C	C	0	0	0	0	0
0	0	1	1	C	1	C	C	0	0	0	4	0
0	0	1	0	2	C	C	C	C	0	0	0	0
0	0	0	1	0	0	0	0	1	0	0	0	0
0	0	0	2	1	C	C	C	0	0	0	0	0
0	0	0	0	1	C	C	1	0	0	0	0	0
0	0	0	0	C	1	1	C	C	0	0	0	0

PROBLEME NO 14 NOMBRE DE SOLUTIONS 135

0	0	0	0	C	C	C	C	C	0	0	0	0	1
1	0	0	0	C	C	C	C	0	0	0	0	1	0
2	0	0	0	C	C	C	C	0	0	0	1	0	0
3	0	0	0	C	C	C	C	0	0	1	0	0	0
4	0	0	0	C	C	C	C	0	1	0	0	0	0
5	0	0	0	C	C	1	C	1	0	0	0	0	0
6	0	0	0	C	C	C	1	C	0	0	0	0	0
7	0	0	0	C	0	1	C	0	0	0	0	0	0
8	0	0	0	C	1	C	C	0	0	0	0	0	0
9	0	0	0	1	C	C	C	0	0	0	0	0	0
10	0	0	1	C	C	C	C	0	0	0	0	0	0
11	0	1	0	C	C	C	C	0	0	0	0	0	0
12	1	0	0	C	C	C	C	0	0	0	0	0	0
14	0	0	0	C	C	C	C	0	0	0	0	0	0
10	2	0	0	C	C	C	C	C	0	0	0	0	0
9	1	1	0	C	C	C	C	0	0	0	0	0	0
8	1	0	1	C	C	C	C	0	0	0	0	0	0
8	3	0	0	C	C	C	C	0	0	0	0	0	0
8	0	2	0	C	C	C	C	0	0	0	0	0	0
7	1	0	0	1	C	C	C	0	0	0	0	0	0
7	2	1	0	C	C	C	C	0	0	0	0	0	0
7	0	1	1	C	C	C	C	0	0	0	0	0	0
6	1	0	0	C	1	C	C	0	0	0	0	0	0
6	2	0	1	C	C	C	C	0	0	0	0	0	0
6	4	0	0	C	C	C	C	0	0	0	0	0	0
6	1	2	0	C	C	0	C	0	0	0	0	0	0
6	0	1	0	1	C	C	C	0	0	0	0	0	0
6	0	0	2	C	C	C	C	0	0	0	0	0	0
5	1	0	0	C	C	1	C	0	0	0	0	0	0
5	2	0	0	1	C	C	C	0	0	0	0	0	0
5	3	1	0	C	C	C	C	0	0	0	0	0	0

1	1	2	0	1	C	C	C	0	0	0	0	0	0
1	1	0	2	C	C	C	C	0	0	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0	0	0
1	1	0	0	1	1	0	0	0	0	0	0	0	0
1	0	1	0	C	C	C	C	0	1	0	0	0	0
1	0	2	0	0	C	C	1	C	0	0	0	0	0
1	0	3	1	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	1	0	0	0	0	0	0	0	0
1	0	0	1	2	C	C	C	0	0	0	0	0	0
1	0	0	2	1	C	C	C	1	0	0	0	0	0
1	0	0	0	1	0	0	1	0	0	0	0	0	0
1	0	0	0	C	1	1	C	0	0	0	0	0	0
0	1	0	0	0	C	C	C	0	0	0	1	0	0
0	2	0	0	0	0	0	0	0	1	0	0	0	0
0	3	0	0	0	0	0	0	0	0	0	0	0	0
0	4	0	0	C	1	C	C	0	0	0	0	0	0
0	5	0	1	C	C	C	C	0	0	0	0	0	0
0	7	0	0	C	C	C	C	0	0	0	0	0	0
0	4	2	0	0	0	0	0	0	0	0	0	0	0
0	3	1	0	1	C	C	C	0	0	0	0	0	0
0	3	0	2	C	C	C	0	0	0	0	0	0	0
0	2	1	0	C	C	1	C	0	0	0	0	0	0
0	2	2	1	0	0	0	0	0	0	0	0	0	0
0	2	0	1	C	1	C	C	0	0	0	0	0	0
0	2	0	0	2	0	0	C	0	0	0	0	0	0
0	1	1	0	C	C	C	C	1	0	0	0	0	0
0	1	2	0	0	1	0	0	0	0	0	0	0	0
0	1	1	0	C	0	C	C	0	0	0	0	0	0
0	1	0	1	C	0	C	1	0	0	0	0	0	0
0	1	0	3	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	C	1	C	0	0	0	0	0	0
0	1	0	0	0	0	2	0	C	0	0	0	0	0
0	0	1	0	C	0	C	C	0	0	1	0	0	0
0	0	2	0	0	0	0	1	0	0	0	0	0	0
0	0	3	0	1	0	C	C	0	0	0	0	0	0
0	0	2	2	0	C	C	0	0	0	0	0	0	0
0	0	1	1	C	C	1	C	0	0	0	0	0	0
0	0	1	0	1	1	0	0	0	0	0	0	0	0
0	0	0	1	0	1	C	C	0	1	0	0	0	0
0	0	0	2	0	1	0	0	0	0	0	0	0	0
0	0	0	1	2	C	C	C	0	0	0	0	0	0
0	0	0	0	1	0	0	0	1	0	0	0	0	0
0	0	0	0	1	1	C	1	0	0	0	0	0	0
0	0	0	0	C	1	C	1	0	0	0	0	0	0
0	0	0	0	0	0	2	0	0	0	0	0	0	0

PROBLEME NO. 15 NOMBRE DE SOLUTIONS 176

0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	C	0	0	0	0	0	0	0	0	1	0
2	0	0	0	C	0	0	C	0	0	0	0	1	0	0
3	0	0	0	C	0	C	C	0	0	0	1	0	0	0
4	0	0	0	0	0	C	0	0	0	1	0	0	0	0
5	0	0	0	0	0	C	0	0	1	0	0	0	0	0
6	0	0	0	C	C	C	C	1	0	0	0	0	0	0
7	0	0	0	C	0	C	1	0	0	0	0	0	0	0

CHAPITRE - IV

CALCUL FORMEL DES COEFFICIENTS D'UN POLYNOME EN LISP 1.5

I - INTRODUCTION

Nous nous proposons d'obtenir ici, grâce à un programme sur machine, les formules donnant les coefficients d'un polynôme en fonction des sommes de puissances semblables des racines de ce polynôme.

Nous montrerons d'abord comment obtenir ces formules puis nous décrivons la façon de traiter ce problème avec le langage LISP 1.5 [6], [7], qui est particulièrement adapté à la manipulation d'expressions littérales.

Il est intéressant de noter que ce langage permet aussi de traiter des problèmes numériques et d'évaluer directement les formules obtenues. Nous pensons que de nombreux algorithmes classiques pourraient, avec profit, faire l'objet d'un traitement identique.

II - PRINCIPE

Nous savons que les relations entre les coefficients a_0, a_1, \dots, a_n du polynôme $P_n(x) = a_0 x^n + a_1 x^{n-1} + \dots + a_{n-1} x + a_n$ et les sommes de puissances semblables de ses racines s'écrivent :

$$(1.2) \quad k \cdot a_k + a_{k-1} S_1 + a_{k-2} S_2 + \dots + a_0 S_k = 0 \quad k = 1, \dots, n.$$

d'où l'expression de a_k en fonction de a_{k-1}, \dots, a_0 et S_k, \dots, S_1 :

$$(2.2) \quad a_k = -\frac{1}{k} [a_{k-1} S_1 + a_{k-2} S_2 + \dots + a_0 S_k] \quad k = 1 \dots n.$$

Ce sont les relations (2.2) que l'on évalue numériquement lorsque l'on veut calculer les coefficients a_i , mais nous désirons exprimer ici chaque coefficient a_k en fonction des seules quantités $a_0, S_1, S_2, \dots, S_k$.

Pour cela il nous "suffit" de remplacer dans l'expression de a_k chacun des coefficients a_1, \dots, a_{k-1} par leur expressions en fonction des seules quantités a_0, S_1, \dots, S_j ($j < k$).

Ainsi de (2.2) nous tirons :

$$\begin{aligned} a_1 &= -a_0 S_1 \\ a_2 &= -\frac{1}{2} [a_1 S_1 + a_0 S_2] = -\frac{a_0}{2} (S_2 - S_1^2) \\ &\text{etc...} \end{aligned}$$

Il est bien évident que ceci serait fastidieux à faire à la main, si nous voulions obtenir les expressions de a_n dès que n devient supérieur à 4 ou 5. C'est donc cet algorithme que nous traiterons en LISP 1.5 et pour la description et l'utilisation de ce langage nous renvoyons le lecteur aux ouvrages et cours spécialisés [6], [7], [8].

III - MISE EN OEUVRE DE L'ALGORITHME

Pour la mise en oeuvre de l'algorithme nous transformons légèrement les formules (2.2) en les réécrivant :

$$a_k = -\frac{a_0}{k} \left[\frac{a_{k-1}}{a_0} S_1 + \frac{a_{k-2}}{a_0} S_2 + \dots + S_k \right] \quad k = 1 \dots n \text{ et}$$

en posant

$$\frac{a_k}{a_0} = b_k \quad k = 1 \dots n \text{ nous aurons :}$$

$$(3.2) \quad b_k = -\frac{1}{k} [b_{k-1}S_1 + b_{k-2}S_2 + \dots + b_1S_{k-1} + S_k] \quad k = 1 \dots n.$$

Nous travaillerons dorénavant sur ces formules.

Soient alors :

a) la liste P formée de :

$$((1.B1)(2.B2) \dots (k.Bk) \dots (n.Bn))$$

b) la liste M formée de :

$$(S_n S_{n-1} \dots S_k \dots S_3 S_2 S_1)$$

c) la liste N qui sera au début de l'algorithme sous la forme :

$$(0 \ 0 \ \dots \ 0 \ \dots \ 0 \ 0 \ 1)$$

Pour obtenir l'expression de (1.B1) nous voyons qu'il suffit de faire :

1°) Le "produit des listes M et N" : c'est-à-dire former la liste

$$((S_n.0)(S_{n-1}.0) \dots (S_k.0) \dots (S_2.0) (S_1.1)).$$

2°) D'éliminer ensuite de la liste obtenue tous les termes du type

$$(S_k.0) \text{ dans lesquels il apparait un } 0.$$

3°) De transformer le terme restant en $(S_1 * 1)$

4°) De calculer la quantité R1 : $(-(S_1 * 1)/1)$ qui est la valeur de b_1 et de conserver ce résultat après l'avoir simplifié dans une liste que nous appellerons RESULTAT.

Ayant ainsi calculé B_1 , nous transformons la liste N en la mettant sous la forme :

$N : (0 \ 0 \ \dots \ 0 \ 1 \ R_1)$ et nous pouvons alors itérer le processus

en formant le produit des listes M et N qui nous donnera B_2 , puis transformer de nouveau la liste N etc...

L'algorithme s'arrêtera lorsque la liste N , complètement modifiée aura la forme $(R_1 \ R_2 \ \dots \ R_k \ \dots \ R_n)$, R_i étant alors l'expression de b_i en fonction des seules quantités S_1, \dots, S_i .

Un certain nombre de remarques s'impose :

* tout d'abord les expressions construites sont complètement parenthésées, la notion de priorité entre opérateurs arithmétiques n'étant pas introduite dans LISP 1.5.

* les opérations arithmétiques élémentaires : somme, différence, produit et quotient ne sont pas définies par le symbole associé à chacune de ces opérations $+, -, *, /$ mais pas le nom des fonctions LISP réalisant ces opérations : PLUS, DIFFERENCE, TIMES, QUOTIENT.

* Si nous désirons évaluer numériquement les expressions littérales obtenues, il faut encore transformer ces expressions en notation préfixée.

Exemple :

L'expression du coefficient B_2 que nous écrivons à la main

$B_2 = -\frac{1}{2} (S_2 - S_1^2)$ sera avec l'algorithme ci-dessus écrite sous la forme :

$E1 : (MINUS((S2 PLUS((MINUS S1)TIMES S1))QUOTIENT 2))$

mais pour l'évaluation numérique de cette expression, il faudra l'écrire sous la forme préfixée.

E2 : (MINUS(QUOTIENT(PLUS S2(TIMES (MINUS S1) S1))2))

Pour rendre cette expression plus lisible nous utiliserons une fonction LISP qui la mettra sous la forme :

E2 : -((S2+((-S1)*S1))/2).

IV - DEVELOPPEMENT ET SIMPLIFICATION DES EXPRESSIONS ARITHMETIQUES

A - DEVELOPPEMENT DES EXPRESSIONS ARITHMETIQUES

Nous entendons par développement des expressions arithmétiques, la distribution des opérations multiplication et quotient par rapport aux opérations somme et différence.

1°) Distribution de l'opération multiplication.

Dans la construction des expressions arithmétiques que nous désirons obtenir, nous sommes amenés à former des expressions arithmétiques du type (TIMES(PLUS(EXPT A 2)B)A). En fait nous ne construirons pas de telles expressions, car nous distribuerons l'opération TIMES par rapport au PLUS pour obtenir l'expression

(PLUS(EXPT A 3) (TIMES B A)).

De la même façon l'expression (TIMES(TIMES A C)C) sera remplacée par (TIMES A (EXPT C 2)).

Ces opérations seront réalisées par la fonction DISTRIM(X ; Y) où X représente le multiplicande et Y le multiplicateur.

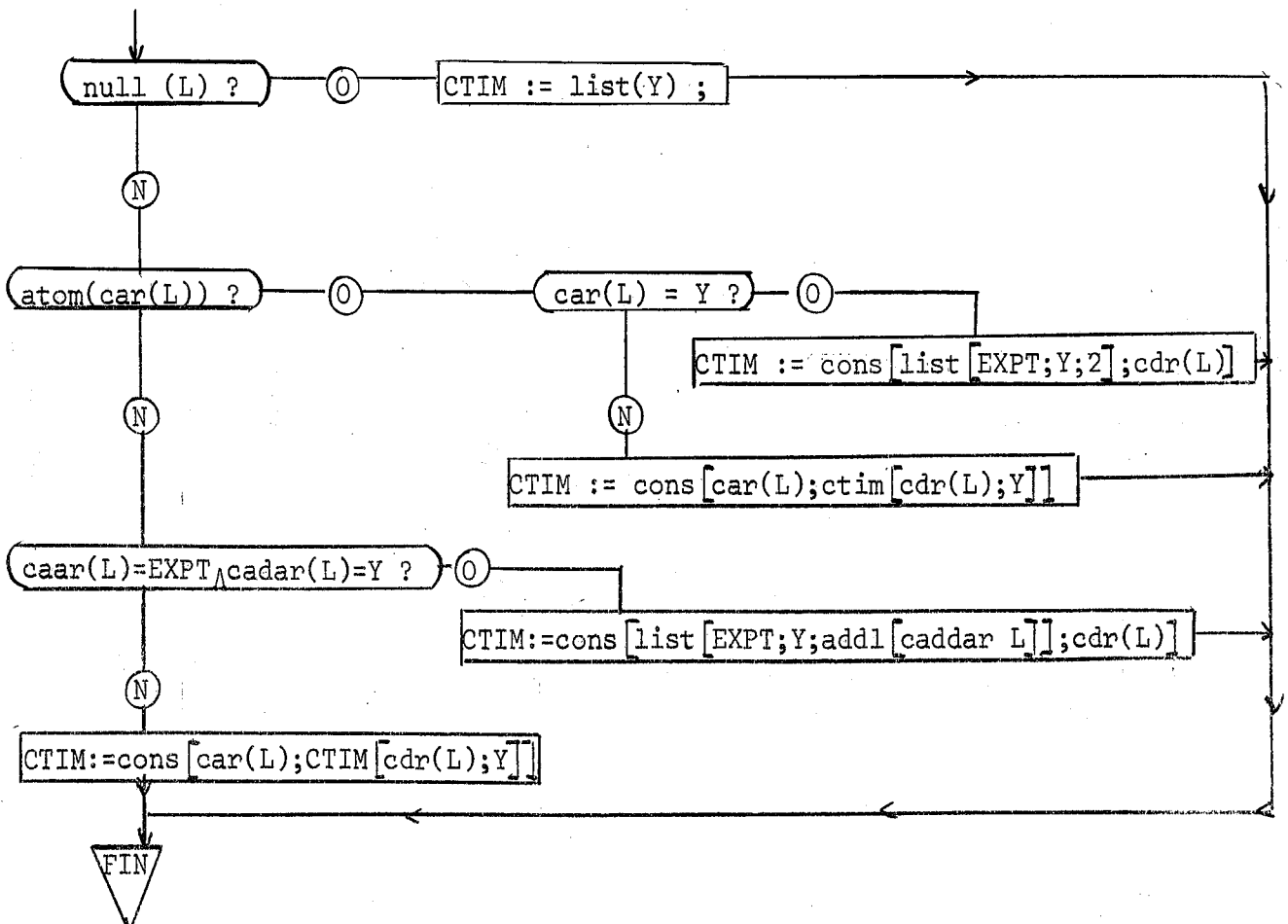
Dans le cas où X est du type

(TIMES A B), nous utiliserons alors une fonction auxiliaire : CTIM(L ; Y)

où L dans l'exemple ci-dessus sera la liste (A B) et dont le but est le suivant :

- si Y appartient à la liste L alors nous construisons la liste (si Y = B par exemple), (A (EXPT B 2)).
- si Y n'appartient pas à la liste L alors nous construisons la liste (A B Y).
- si L contient une sous liste du type (EXPT Y N) alors nous construisons la liste : (A (EXPT Y (N+1))).

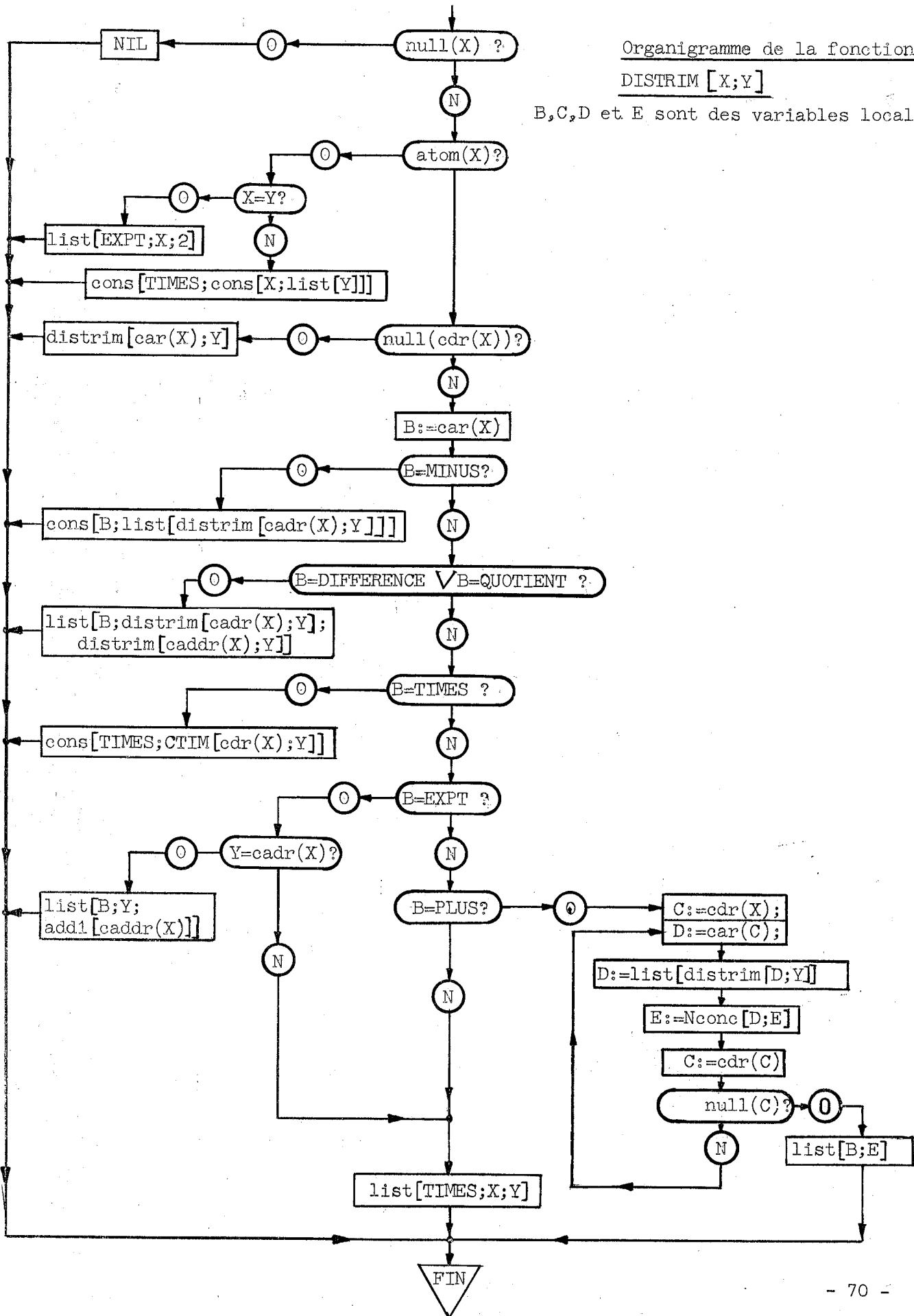
Voici l'organigramme de ces 2 fonctions :



Organigramme de la fonction

DISTRIM [X;Y]

B,C,D et E sont des variables locales



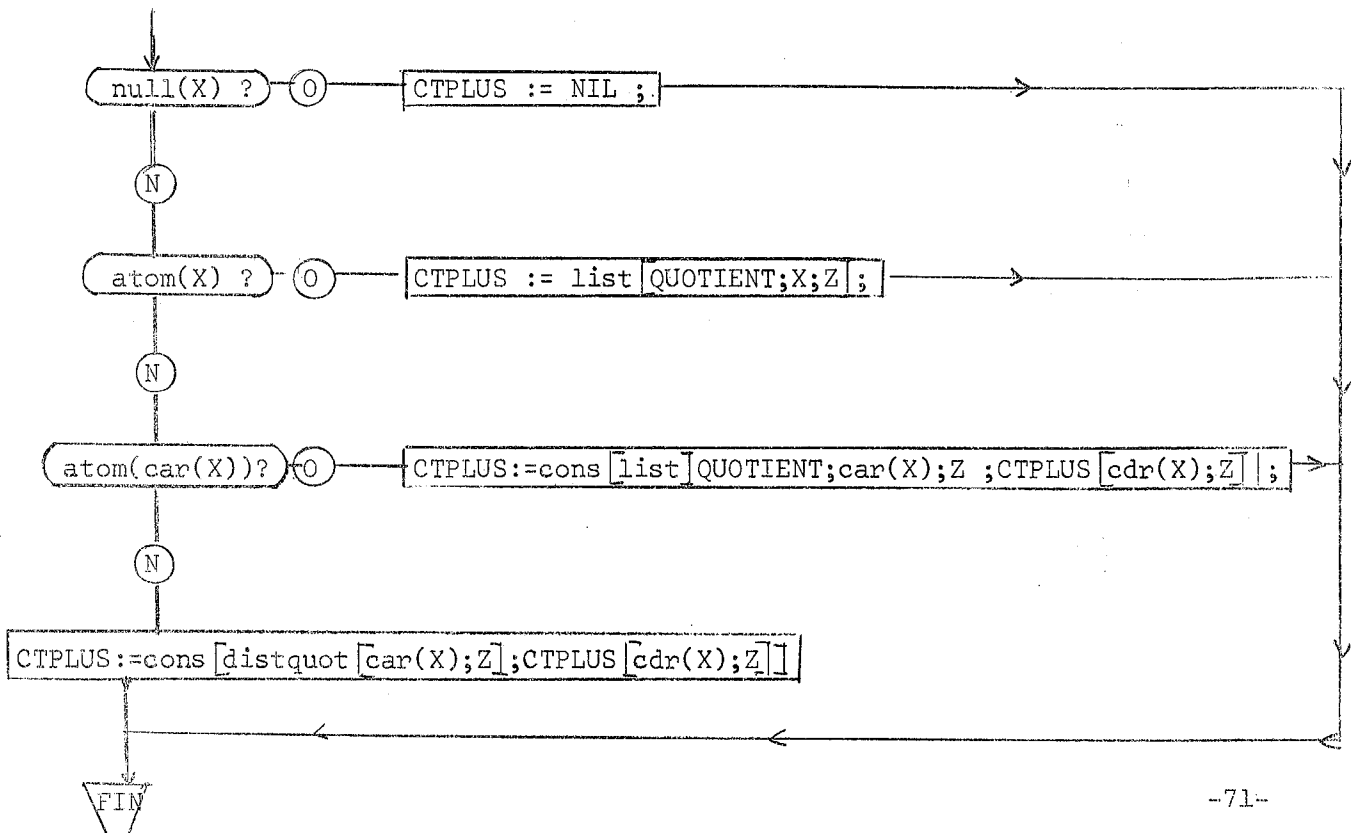
2°) Distribution de l'opération division

De la même façon que nous avons distribué l'opération multiplication par rapport à l'addition, nous distribuerons l'opération division : cette opération a lieu lorsqu'on a construit l'expression arithmétique correspondant à $k.b_k$ et que l'on veut en déduire b_k . Ceci est réalisé par la fonction $\text{Distquot}[L;Z]$ où L représente le dividende, le diviseur étant Z.

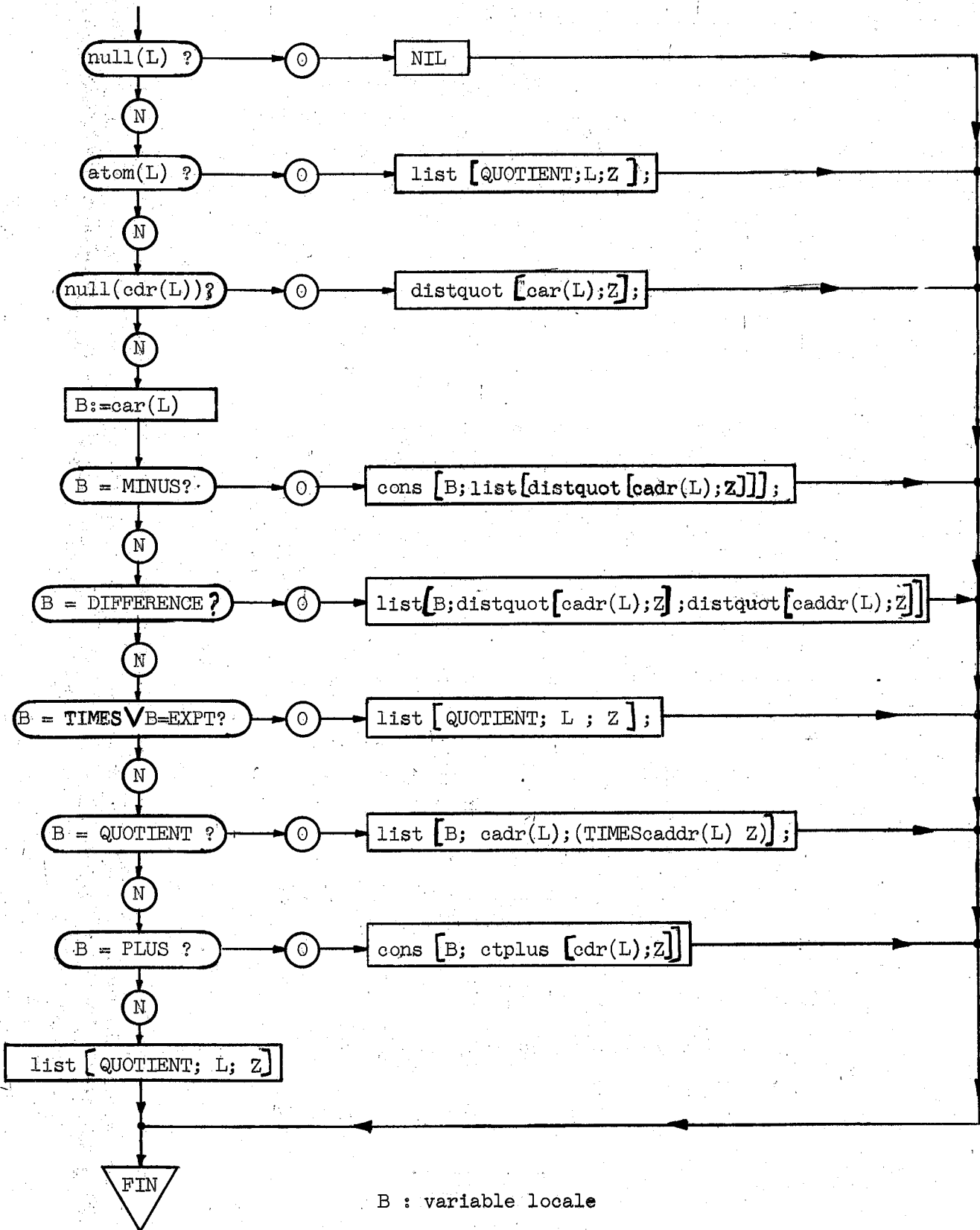
Remarquons que Z est un entier dans notre cas et que lorsque L est du type $(\text{QUOTIENT}(\text{TIMES A B}) 2)$ et que $Z = 3$ nous construirons la liste $(\text{QUOTIENT}(\text{TIMES A B})6)$.

Enfin dans le cas où l'opérateur de la liste L est l'opérateur PLUS, nous utiliserons la fonction CTPLUS pour distribuer la division par rapport à la somme. Ainsi :

soit $L = (\text{PLUS}(\text{TIMES A B})C)$ et soit $Z = 2$ nous utiliserons $\text{CTPLUS}[\text{cdr}(L);2]$.

Organigramme de la fonction CTPLUS [X;Z].

Organigramme de la fonction DISTQUOT [L;Z];



B - REGLES DE SIMPLIFICATION

Les règles de simplification que nous avons utilisées dans la réécriture des expressions arithmétiques préfixées sous forme normale sont les suivantes :

Nous avons défini un ordre de priorité pour les fonctions arithmétiques LISP telles que PLUS, TIMES etc...

Ces ordres de priorité sont les suivants :

FONCTION	Ordre de priorité
MINUS (moins unitaire)	0
PLUS	1
DIFFERENCE	2
TIMES (multiplication)	3
QUOTIENT	3
EXPT (exponentielle)	4
Autres fonctions : SINUS, COSINUS etc	5

Une expression arithmétique complètement parenthésée ayant une structure de liste, nous allons modifier la structure de cette liste :

1°) en transformant la suite :

<opérateur 1 opérande 1 opérande 2>

en

<opérande 1 opérateur 1 opérande 2>

2°) en éliminant, grâce aux ordres de priorité que nous avons défini, les

parenthèses inutiles : si opérande 1 \equiv <opérateur 2 opérande 3

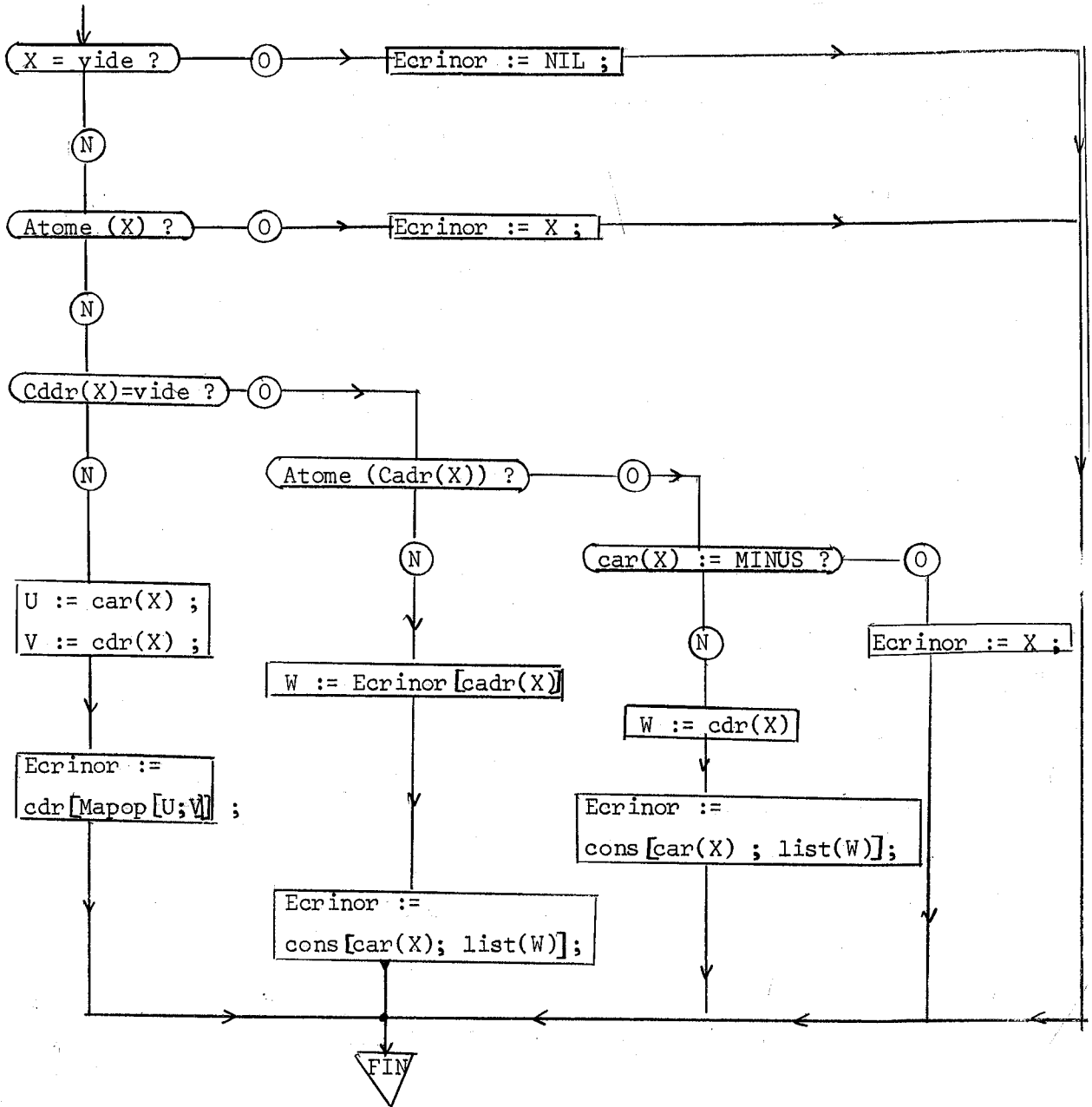
opérande 4> et si la priorité de opérande 2 > priorité de opérande 1

nous construirons la suite :

<opérande 3 opérateur 2 opérande 4 opérateur 1 opérande 2>

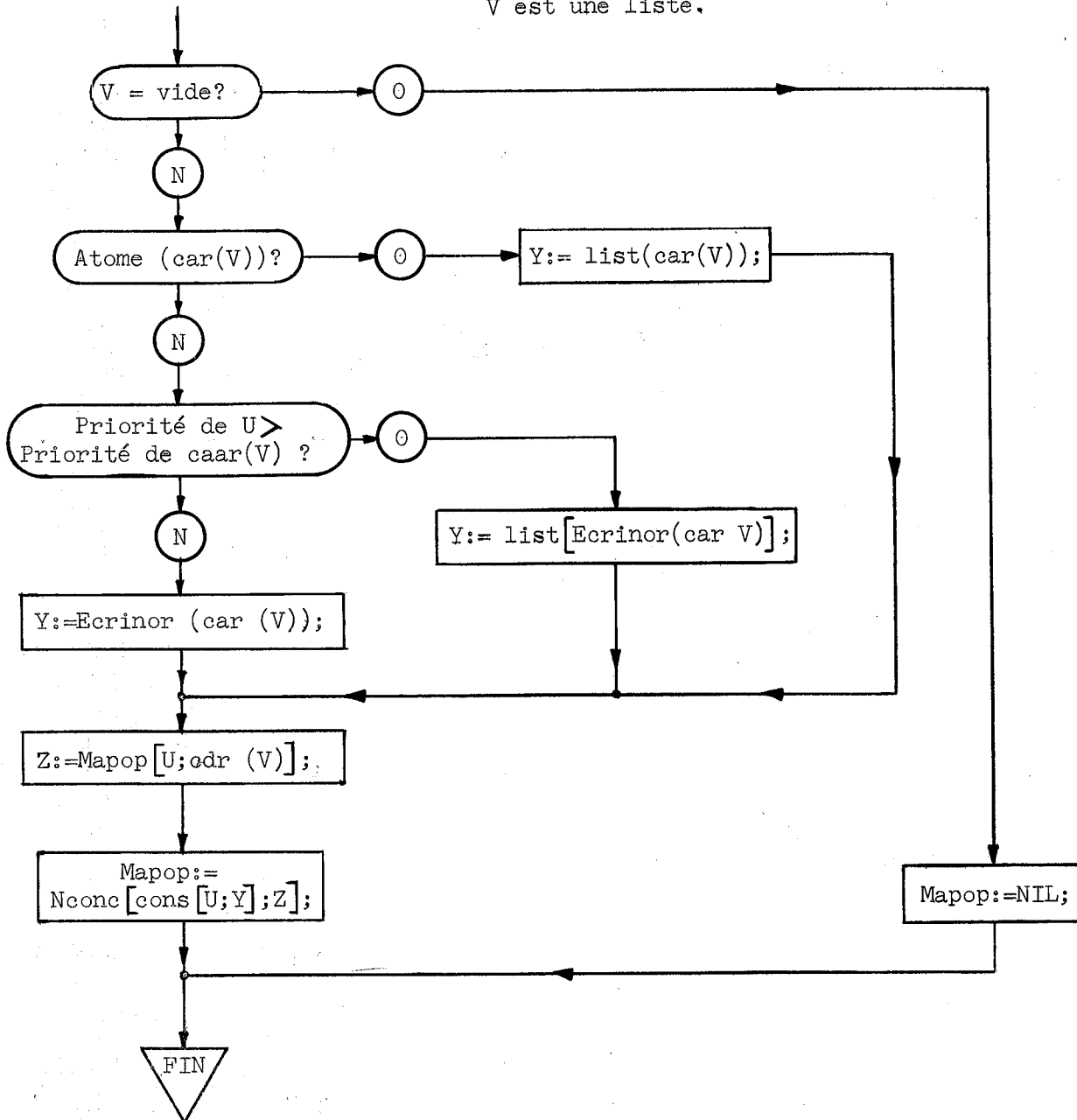
Ces 2 opérations se feront simultanément à l'aide de 2 fonctions LISP récursives :
Ecrinor(X) et Mapop(u;v) dont voici les organigrammes :

Organigramme de la fonction ECRINOR X représente la liste préfixée.



Organigramme de la fonction $MAPOP [U;V];$

U est une fonction arithmétique (PLUS, TIMES, etc...)
 V est une liste.



Voici 2 exemples de réécriture d'expression arithmétiques préfixées à l'aide de ces 2 fonctions.

Exemple 1 :

Soit l'expression arithmétique préfixée : $X := (\text{TIMES } A \ B)$. Décrivons $\text{Ecrinor}(X)$. (cf. organigramme).

- X n'est pas vide
 - X n'est pas un atome
 - Le $\text{Cddr}(X) = (B)$ n'est pas vide
 donc $U := \text{cdr}(X)$ soit $U := \text{TIMES}$
 et $V := \text{cdr}(X)$ soit $V := (A \ B)$
 puis $\text{Ecrinor} := \text{cdr}[\text{Mapop}[U;V]];$

$\text{Mapop}[U;V]$ avec $U := \text{TIMES}$ et $V := (A \ B)$

- V n'est pas vide
 - $\text{car}(V) = A$ est un atome donc
 $Y := \text{list}(\text{car}(V))$ soit $Y := (A)$;
 puis $Z := \text{mapop}[\text{TIMES} ; (B)]$ soit
 $Z := (\text{TIMES } B)$
 et enfin $\text{Mapop} := \text{nconc}[\text{cons}[\text{TIMES};(A)]; (\text{TIMES } B)];$
 soit $\text{Mapop} := (\text{TIMES } A \ \text{TIMES } B)$
 d'ou $\text{Ecrinor} := \text{cdr}[(\text{TIMES } A \ \text{TIMES } B)];$
 soit $\text{Ecrinor} := (A \ \text{TIMES } B) ;$

Exemple 2 :

Soit $X := (\text{PLUS } (\text{TIMES } A \ B)C)$; décrivons $\text{Ecrinor}(X)$;

- X n'est pas vide
 - X n'est pas un atome
 - le $\text{cddr}(X) = (C)$ n'est pas vide
 donc $U := \text{PLUS}$
 $V := ((\text{TIMES } A \ B)C)$
 et $\text{Ecrinor} := \text{cdr}[\text{Mapop}[U;V]];$

Mapop[U;V] avec U := PLUS et V := ((TIMES A B)C)

- V n'est pas vide

- car(V) n'est pas un atome (car(V)=(TIMES A B))

- la priorité de U n'est pas plus grande que celle du caar(V)

caar(V) = TIMES et

priorité de PLUS = 1 < priorité de TIMES = 3

donc Y := Ecrinor[car(V)]

soit Y := Ecrinor[(TIMES A B)];

d'où Y := (A TIMES B) (cf. exemple 1).

puis :

Z := Mapop[PLUS ;(C)];

soit Z := (PLUS C) ;

et enfin

Mapop := (PLUS A TIMES B PLUS C)

d'où finalement

Ecrinor := (A TIMES B PLUS C).

Sortie des résultats

L'expression arithmétique ainsi simplifiée est enfin modifiée pour la sortie des résultats : les noms des fonctions arithmétiques sont remplacés par les symboles des opérations arithmétiques correspondantes et le résultat obtenu est soit imprimé, soit perforé selon ce que l'utilisateur veut en faire.

Remarquons ici que les expressions arithmétiques obtenues sur cartes perforées peuvent être directement introduites dans un programme ALGOL et affectées à des variables de ce programme pour être évaluées.

COEFFICIENT NO 1

-S1

COEFFICIENT NO 2

 $S1^{**2/2} - S2/2$

COEFFICIENT NO 3

 $S1*S2/2 - (S3/3 + S1^{**3/6})$

COEFFICIENT NO 4

 $(S2^{**2/8} + S1*S3/3 + S1^{**4/24}) - (S4/4 + S1^{**2}*S2/4)$

COEFFICIENT NO 5

 $(S2*S3/6 + S1*S4/4 + S1^{**3}*S2/12) - (S5/5 + S1*S2^{**2/8} + S1^{**2}*S3/6 + S1^{**5}/120)$

COEFFICIENT NO 6

 $(S3^{**2/18} + S2*S4/8 + S1*S5/5 + S1^{**2}*S2^{**2/16} + S1^{**3}*S3/18 + S1^{**6}/720) - (S6/6 + S2^{**3}/48 + S1*S2*S3/6 + S1^{**2}*S4/8 + S1^{**4}*S2/48)$

COEFFICIENT NO 7

 $(S2*S4/12 + S2*S5/10 + S1*S6/6 + S1*S2^{**3}/48 + S1^{**2}*S2*S3/12 + S1^{**3}*S4/24 + S1^{**5}*S2/240) - (S7/7 + S2^{**2}*S3/24 + S1*S3^{**2/18} + S1*S2*S4/8 + S1^{**2}*S5/10 + S1^{**3}*S2^{**2/48} + S1^{**4}*S3/72 + S1^{**7}/5040)$

COEFFICIENT NO 8

 $(S4^{**2/32} + S3*S5/15 + S2*S6/12 + S2^{**4}/384 + S1*S7/7 + S1*S2^{**2}*S3/24 + S1^{**2}*S3^{**2/36} + S1^{**2}*S2*S4/16 + S1^{**3}*S5/30 + S1^{**4}*S2^{**2/192} + S1^{**5}*S3/360 + S1^{**8}/40320) - (S8/8 + S2*S3^{**2/36} + S2^{**2}*S4/32 + S1*S3*S4/12 + S1*S2*S5/10 + S1^{**2}*S6/12 + S1^{**2}*S2^{**3/96} + S1^{**3}*S2*S3/36 + S1^{**4}*S4/96 + S1^{**6}*S2/1440)$

COEFFICIENT NO 9

 $(S4*S5/20 + S3*S6/18 + S2*S7/14 + S2^{**3}*S3/144 + S1*S8/8 + S1*S2*S3^{**2/36} + S1*S2^{**2}*S4/32 + S1^{**2}*S3*S4/24 + S1^{**2}*S2*S5/20 + S1^{**3}*S6/36 + S1^{**3}*S2^{**3/288} + S1^{**4}*S2*S3/144 + S1^{**5}*S4/480 + S1^{**7}*S2/10080) - (S9/9 + S3^{**3}/162 + S2*S3*S4/24 + S2^{**2}*S5/40 + S1*S4^{**2/32} + S1*S3*S5/15 + S1*S2*S6/12 + S1*S2^{**4}/384 + S1^{**2}*S7/14 + S1^{**2}*S2^{**2}*S3/48 + S1^{**3}*S3^{**2/108} + S1^{**3}*S2*S4/48 + S1^{**4}*S5/120 + S1^{**5}*S2^{**2/960} + S1^{**6}*S3/2160 + S1^{**9}/362880)$

COEFFICIENT NO 10

$(S5^{**}2/50 + S4^{*}S6/24 + S3^{*}S7/21 + S2^{*}S8/16 + S2^{**}2^{*}S3^{**}2/144$
 $+ S2^{**}3^{*}S4/192 + S1^{*}S9/9 + S1^{*}S3^{**}3/162 + S1^{*}S2^{*}S3^{*}S4/24 +$
 $S1^{*}S2^{**}2^{*}S5/40 + S1^{**}2^{*}S4^{**}2/64 + S1^{**}2^{*}S3^{*}S5/30 + S1^{**}2^{*}S2^{*}S6/24$
 $+ S1^{**}2^{*}S2^{**}4/768 + S1^{**}3^{*}S7/42 + S1^{**}3^{*}S2^{**}2^{*}S3/144 + S1^{**}4^{*}S3^{**}2/432$
 $+ S1^{**}4^{*}S2^{*}S4/192 + S1^{**}5^{*}S5/600 + S1^{**}6^{*}S2^{**}2/5760 + S1^{**}7^{*}S3/15120$
 $+ S1^{**}10/3628800) - (S10/10 + S3^{**}2^{*}S4/72 + S2^{*}S4^{**}2/64 + S2^{*}S3^{*}S5/30$
 $+ S2^{**}2^{*}S6/48 + S2^{**}5/3840 + S1^{*}S4^{*}S5/20 + S1^{*}S3^{*}S6/18 + S1^{*}S2^{*}S7/14$
 $+ S1^{*}S2^{**}3^{*}S3/144 + S1^{**}2^{*}S8/16 + S1^{**}2^{*}S2^{*}S3^{**}2/72 + S1^{**}2^{*}S2^{**}2^{*}S4/64$
 $+ S1^{**}3^{*}S3^{*}S4/72 + S1^{**}3^{*}S2^{*}S5/60 + S1^{**}4^{*}S6/144 + S1^{**}4^{*}S2^{**}3/1152$
 $+ S1^{**}5^{*}S2^{*}S3/720 + S1^{**}6^{*}S4/2880 + S1^{**}8^{*}S2/80640)$

COEFFICIENT NO 11

$(S5^{*}S6/30 + S4^{*}S7/28 + S3^{*}S8/24 + S2^{*}S9/18 + S2^{*}S3^{**}3/324$
 $+ S2^{**}2^{*}S3^{*}S4/96 + S2^{**}3^{*}S5/240 + S1^{*}S10/10 + S1^{*}S3^{**}2^{*}S4/72$
 $+ S1^{*}S2^{*}S4^{**}2/64 + S1^{*}S2^{*}S3^{*}S5/30 + S1^{*}S2^{**}2^{*}S6/48 + S1^{*}S2^{**}5/3840$
 $+ S1^{**}2^{*}S4^{*}S5/40 + S1^{**}2^{*}S3^{*}S6/36 + S1^{**}2^{*}S2^{*}S7/28 + S1^{**}2^{*}S2^{**}3^{*}S3/288$
 $+ S1^{**}3^{*}S8/48 + S1^{**}3^{*}S2^{*}S3^{**}2/216 + S1^{**}3^{*}S2^{**}2^{*}S4/192 + S1^{**}4^{*}S3^{*}S4/288$
 $+ S1^{**}4^{*}S2^{*}S5/240 + S1^{**}5^{*}S6/720 + S1^{**}5^{*}S2^{**}3/5760 + S1^{**}6^{*}S2^{*}S3/4320$
 $+ S1^{**}7^{*}S4/20160 + S1^{**}9^{*}S2/725760) - (S11/11 + S3^{*}S4^{**}2/96 + S3^{**}2^{*}S5/90$
 $+ S2^{*}S4^{*}S5/40 + S2^{*}S3^{*}S6/36 + S2^{**}2^{*}S7/56 + S2^{**}4^{*}S3/1152 +$
 $S1^{*}S5^{**}2/50 + S1^{*}S4^{*}S6/24 + S1^{*}S3^{*}S7/21 + S1^{*}S2^{*}S8/16 + S1^{*}S2^{**}2^{*}S3^{**}2/144$
 $+ S1^{*}S2^{**}3^{*}S4/192 + S1^{**}2^{*}S9/18 + S1^{**}2^{*}S3^{**}3/324 + S1^{**}2^{*}S2^{*}S3^{*}S4/48$
 $+ S1^{**}2^{*}S2^{**}2^{*}S5/80 + S1^{**}3^{*}S4^{**}2/192 + S1^{**}3^{*}S3^{*}S5/90 + S1^{**}3^{*}S2^{*}S6/72$
 $+ S1^{**}3^{*}S2^{**}4/2304 + S1^{**}4^{*}S7/168 + S1^{**}4^{*}S2^{**}2^{*}S3/576 + S1^{**}5^{*}S3^{**}2/2160$
 $+ S1^{**}5^{*}S2^{*}S4/960 + S1^{**}6^{*}S5/3600 + S1^{**}7^{*}S2^{**}2/40320 + S1^{**}8^{*}S3/120960$
 $+ S1^{**}11/39916800)$

COEFFICIENT NO 12

$S6^{**}2/72 + S5^{*}S7/35 + S4^{*}S8/32 + S3^{*}S9/27 + S3^{**}4/1944 +$
 $S2^{*}S10/20 + S2^{*}S3^{**}2^{*}S4/144 + S2^{**}2^{*}S4^{**}2/256 + S2^{**}2^{*}S3^{*}S5/120$
 $+ S2^{**}3^{*}S6/288 + S2^{**}6/46080 + S1^{*}S11/11 + S1^{*}S3^{*}S4^{**}2/96 + S1^{*}S3^{**}2^{*}S5/90$
 $+ S1^{*}S2^{*}S4^{*}S5/40 + S1^{*}S2^{*}S3^{*}S6/36 + S1^{*}S2^{**}2^{*}S7/56 + S1^{*}S2^{**}4^{*}S3/1152$
 $+ S1^{**}2^{*}S5^{**}2/100 + S1^{**}2^{*}S4^{*}S6/48 + S1^{**}2^{*}S3^{*}S7/42 + S1^{**}2^{*}S2^{*}S8/32$
 $+ S1^{**}2^{*}S2^{**}2^{*}S3^{**}2/288 + S1^{**}2^{*}S2^{**}2^{*}S4/384 + S1^{**}3^{*}S9/54 +$
 $S1^{**}3^{*}S3^{**}3/972 + S1^{**}3^{*}S2^{*}S3^{*}S4/144 + S1^{**}3^{*}S2^{**}2^{*}S5/240 + S1^{**}4^{*}S4^{**}2/768$
 $+ S1^{**}4^{*}S3^{*}S5/360 + S1^{**}4^{*}S2^{*}S6/288 + S1^{**}4^{*}S2^{**}4/9216 + S1^{**}5^{*}S7/840$
 $+ S1^{**}5^{*}S2^{**}2^{*}S3/2880 + S1^{**}6^{*}S3^{**}2/12960 + S1^{**}6^{*}S2^{*}S4/5760 + S1^{**}7^{*}S5/25200$
 $+ S1^{**}8^{*}S2^{**}2/322560 + S1^{**}9^{*}S3/1068640 + S1^{**}12/479001600) - (S12/12 + S4^{**}3/38$
 $+ S3^{*}S4^{*}S5/60 + S3^{**}2^{*}S6/108 + S2^{*}S5^{**}2/100 + S2^{*}S4^{*}S6/48 +$
 $S2^{*}S3^{*}S7/42 + S2^{**}2^{*}S8/64 + S2^{**}3^{*}S3^{**}2/864 + S2^{**}4^{*}S4/1536 +$
 $S1^{*}S5^{*}S6/30 + S1^{*}S4^{*}S7/28 + S1^{*}S3^{*}S8/24 + S1^{*}S2^{*}S9/18 + S1^{*}S2^{*}S3^{**}3/324$
 $+ S1^{*}S2^{**}2^{*}S3^{*}S4/96 + S1^{*}S2^{**}3^{*}S5/240 + S1^{**}2^{*}S10/20 + S1^{**}2^{*}S3^{**}2^{*}S4/144$
 $+ S1^{**}2^{*}S2^{*}S4^{**}2/128 + S1^{**}2^{*}S2^{*}S3^{*}S5/60 + S1^{**}2^{*}S2^{**}2^{*}S6/96$
 $+ S1^{**}2^{*}S2^{**}5/7680 + S1^{**}3^{*}S4^{*}S5/120 + S1^{**}3^{*}S3^{*}S6/108 + S1^{**}3^{*}S2^{*}S7/84$
 $+ S1^{**}3^{*}S2^{**}3^{*}S3/864 + S1^{**}4^{*}S8/192 + S1^{**}4^{*}S2^{*}S3^{**}2/864 + S1^{**}4^{*}S2^{**}2^{*}S4/768$
 $+ S1^{**}5^{*}S3^{*}S4/1440 + S1^{**}5^{*}S2^{*}S5/1200 + S1^{**}6^{*}S6/4320 + S1^{**}6^{*}S2^{**}3/34560$
 $+ S1^{**}7^{*}S2^{*}S3/30240 + S1^{**}8^{*}S4/161280 + S1^{**}10^{*}S2/7257600)$

CHAPITRE - V

CALCUL DES COEFFICIENTS DU POLYNOME CARACTERISTIQUE D'UNE MATRICE A

I - INTRODUCTION

Nous voulons dans ce chapitre utiliser les résultats obtenus dans le chapitre 3 pour calculer les coefficients du polynôme caractéristique d'une matrice $A \in \mathcal{M}_{n,n}(\mathbb{R})$ et comparer les résultats obtenus en utilisant d'une part la méthode de Leverrier et d'autre part les expressions littérales du paragraphe 3 du chapitre 3.

Nous exposerons dans une première partie le principe de la méthode de Leverrier, la deuxième partie étant consacrée à la mise en oeuvre des deux méthodes et à leur comparaison sur des résultats expérimentaux.

II - METHODE DE LEVERRIER [9]

Soit une matrice $A \in \mathcal{M}_{n,n}(\mathbb{R})$ et soit :

$F(\mu) = \det(A - \mu I)$ le polynôme caractéristique de cette matrice.

Soient $\mu_1, \mu_2, \dots, \mu_n$ les racines de $F(\mu) = 0$. Si on peut connaître les sommes :

$$S_1 = \mu_1 + \mu_2 + \dots + \mu_n$$

$$S_2 = \mu_1^2 + \mu_2^2 + \dots + \mu_n^2$$

$$\vdots$$

$$S_n = \mu_1^n + \mu_2^n + \dots + \mu_n^n$$

On en déduira de par les relations de Newton (cha. III, §1) les coefficients du polynôme $F(\mu)$, le coefficient p_0 de μ^n pouvant être pris égal à $(-1)^n$.

Or d'après l'expression de $\det(A - I)$:

$$\det(A - \mu I) = (-1)^n [\mu^n - (a_{11} + a_{22} + \dots + a_{nn})\mu^{n-1} + \dots]$$

Nous avons $S_1 = \mu_1 + \mu_2 + \dots + \mu_n = a_{11} + a_{22} + \dots + a_{nn} = \text{Trace}(A)$.

Mais si A admet pour valeurs propres $\mu_1, \mu_2, \dots, \mu_n$

A^2 admettra pour valeurs propres $\mu_1^2, \mu_2^2, \dots, \mu_n^2$

$$\vdots$$

A^n admettra pour valeurs propres $\mu_1^n, \mu_2^n, \dots, \mu_n^n$

D'un : $S_k = \text{Tr}(A^k) \quad k = 1 \dots n$.

Le calcul des coefficients p_k se fait alors par :

$$(1.2) \quad p_k/p_0 = -\frac{1}{k} [p_{k-1}/p_0 S_1 + \dots + p_1/p_0 S_{k-1} + S_k]$$

en commençant par $p_1/p_0 = -S_1$ etc...

III - COMPARAISON AVEC LES EXPRESSIONS LITTÉRALES

Rappelons que les expressions donnant les coefficients d'un polynôme en fonction des sommes des puissances semblables de ses racines s'écrivent :

$$(1.3) \quad p_i/p_0 = \sum \frac{(-1)^{\lambda_1 + \lambda_2 + \dots + \lambda_i}}{\lambda_1! \lambda_2! \dots \lambda_i!} S_1^{\lambda_1} S_2^{\lambda_2} \dots S_i^{\lambda_i}$$

la somme Σ s'étendant à tous les i -uples $(\lambda_1, \lambda_2, \dots, \lambda_i)$ d'entiers positifs ou nuls satisfaisant à :

$$1.\lambda_1 + 2.\lambda_2 + \dots + i.\lambda_i = i,$$

ces formules pouvant être considérées comme le résultat de l'inversion formelle des relations de Newton.

Ayant calculé les i -uples solutions de l'équation $1.\lambda_1 + \dots + i.\lambda_i = i$ nous avons écrit un programme Algol permettant de calculer les coefficients p_i du polynôme caractéristique d'une matrice A à l'aide des formules (1.2) (Méthode de Leverrier) et des formules (1.3) (calcul formel).

Dans le second cas, afin de diminuer les erreurs machines dues à des différences successives de nombres opposés voisins en valeur absolue, le calcul de chaque coefficient p_i s'est effectué de la manière suivante : nous avons accumulé séparément toutes les quantités positives S_p et toutes les quantités négatives S_N intervenant dans la somme permettant de calculer p_i et nous avons ensuite réalisé la somme $S_p + S_N$. Voici les résultats obtenus sur deux exemples :

Exemple 1 : Soit la matrice d'ordre 5 :

$$A = \begin{vmatrix} 5 & 2.5 & 1.5 & 0.5 & -0.5 \\ 3 & 5.5 & 0.5 & -0.5 & -1.5 \\ 7/3 & 5/6 & 89/15 & -7/6 & 13/6 \\ 5/3 & 5/30 & -5/6 & 37/6 & 17/6 \\ 1 & -0.5 & -1.5 & -2.5 & 6.5 \end{vmatrix}$$

Coefficients du polynôme caractéristique rangés dans l'ordre des puissances décroissantes.

Coefficients exacts	Méthode de Leverrier	Calcul formel
-1	-1	-1
29	28.99999	28.999999
-312	-311.99999	-311.99999
1516	1515.9999	1515.9999
-3152	-3151.9996	-3151.9995
1920	+1919.9984	1920.0078

Norme euclidienne de l'erreur : $0.16163922 \cdot 10^{-2}$ $0.7828704 \cdot 10^{-2}$

Erreur relative : $0.40387161 \cdot 10^{-6}$ $0.19560793 \cdot 10^{-5}$

Norme du max de l'erreur : $0.15716553 \cdot 10^{-2}$ $0.78124999 \cdot 10^{-2}$

Exemple 2 : Soit la matrice A d'ordre 7 :

-0.778	2	1.778	1.333	1.111	0.889	0.667
1.882	-4	1.378	0.933	0.711	0.489	0.267
1.422	1.2	-0.022	0.533	0.311	0.089	-0.133
0.622	0.4	0.178	0.733	-0.489	-0.711	-0.933
0.222	0	-0.222	-0.667	1.111	-1.111	-1.333
-0.178	-0.4	-0.622	-1.067	-1.289	1.489	-1.733
-0.578	-0.8	-1.022	-1.467	-1.689	-1.911	1.867

Les résultats obtenus sont les suivants (les coefficients sont rangés dans l'ordre des puissances décroissantes :

Coefficients exacts	Méthode de Leverrier	Calcul formel
1	-1	-1
4	4	4
14	13.999999	13.999999
-56	-55.999995	-55.999996
-49	-48.999989	-48.999988
196	195.99995	195.99997
36	35.999969	36
-144	-144.00003	-144

Norme euclidienne de l'erreur : $0.65876482 \cdot 10^{-4}$ $0.32836912 \cdot 10^{-4}$

Erreur relative : $0.25604008 \cdot 10^{-6}$ $0.12762621 \cdot 10^{-6}$

Norme du max de l'erreur : $0.45776367 \cdot 10^{-4}$ $0.30517578 \cdot 10^{-4}$

En conclusion il semble donc que les deux méthodes sont sensiblement équivalentes : nous aurions pu penser que la méthode utilisant l'expression exacte des coefficients p_i en fonction des sommes des puissances semblables devait donner de meilleurs résultats que la méthode de Leverrier. En fait il n'en est rien et cela semble surtout dû à ce que le nombre d'opérations dans le calcul formel est beaucoup plus élevé.

Le calcul des sommes de puissances semblables étant le même dans les deux cas il nous faut, par exemple pour $n=8$, faire encore une cinquantaine d'additions et de multiplications pour obtenir tous les p_i dans la méthode de Leverrier, alors que l'utilisation des expressions formelles en exige plus de deux cents.

CHAPITRE - VI

CALCUL DES COEFFICIENTS DU POLYNOME CARACTERISTIQUE ET MATRICE ANTISCALAIRES

I - INTRODUCTION

Soit A une matrice $\in \mathcal{M}_{n,n}(\mathbb{R})$ et soit

$$F_A(\lambda) = \det(A - \lambda I) = (-1)^n \lambda^n + p_{n-1} \lambda^{n-1} + \dots + p_1 \lambda + p_0$$

son polynôme caractéristique.

Soit z le vecteur de \mathbb{R}^n de composantes :

$$z = \begin{pmatrix} p_{n-1} \\ p_{n-2} \\ \vdots \\ p_1 \\ p_0 \end{pmatrix}$$

Si nous calculons numériquement les coefficients du polynôme caractéristique de A par une méthode quelconque (Leverrier, Souriau, Danilewsky, ...), nous obtiendrons un vecteur \tilde{z} qui, du fait de l'accumulation des erreurs de calcul en machine est différent de z : $\tilde{z} \neq z$.

Nous nous proposons alors de savoir si le vecteur \tilde{z} ne peut pas être considéré comme le vecteur exact des coefficients du polynôme caractéristique de la matrice $A + \delta A$, δA pouvant être assimilée à une matrice de perturbations de la matrice A : notre problème est alors de déterminer la matrice δA .

Pour que ceci soit possible, il nous faut connaître les relations existant entre les coefficients p_i $i=0, \dots, n-1$ du polynôme caractéristique de A et ceux, p'_i $i=0, \dots, n-1$, du polynôme caractéristique de $A+\delta A$. Nous utiliserons donc les résultats de Duc-Jacquet [3] sur les matrices "augmentées".

II - DETERMINATION DE δA SOUS LA FORME uv^T , u et $v \in \mathbb{R}^n$

Soit $A \in \mathcal{M}_{n,n}(\mathbb{R})$ et u et v deux vecteurs de \mathbb{R}^n .

Nous savons que si

$$F_A(\lambda) = \det(A - \lambda I) = (-1)^n \lambda^n + p_{n-1} \lambda^{n-1} + \dots + p_1 \lambda + p_0$$

alors :

$$F_{A+uv^T}(\lambda) = \det(A+uv^T - \lambda I) = F_A(\lambda) + v^T f(\lambda) \text{ où :}$$

$$\left| \begin{array}{l} f(\lambda) = \lambda^{n-1} \varphi_{n-1} + \lambda^{n-2} \varphi_{n-2} + \dots + \lambda \varphi_1 + \varphi_0 \quad \varphi_i \in \mathbb{R}^n \\ \varphi_{n-1} = (-1)^{n-1} u \\ \varphi_{i-1} = A \varphi_i - p_i u \quad i = n-1, \dots, 1. \end{array} \right.$$

Si $F_{A+uv^T}(\lambda) = (-1)^n \lambda^n + p'_{n-1} \lambda^{n-1} + \dots + p'_1 \lambda + p'_0$ on a alors :

$$p'_{n-1} = p_{n-1} + v^T \varphi_{n-1}$$

$$\underline{p'_i} = p_i + v^T \varphi_i \quad i = n-1, \dots, 0$$

Nous nous proposons donc de déterminer $u, v \in \mathbb{R}^n$ tel que \tilde{z} obtenu numériquement à partir de la matrice A , soit le vecteur exact des coefficients du polynôme caractéristique de $A+uv^T$.

Si de tels vecteurs existent ils devront naturellement vérifier les relations :

$$\begin{aligned}
 \text{(a)} \quad \varphi_{n-1} &= (-1)^{n-1} u \\
 \text{(b)} \quad \varphi_{i-1} &= A \varphi_i - p_i u \quad i=n-1, \dots, 1 \\
 \text{(c)} \quad p_i' - p_i &= v^T \varphi_i \quad i=n-1, \dots, 0.
 \end{aligned}$$

Nous supposerons alors que $p_i' - p_i = v^T \varphi_i = r_i$ est connu pour $i = n-1, \dots, 0$.

En substituant (a) et (b) dans les n relations (c) nous obtenons alors le système linéaire suivant :

$$\begin{bmatrix}
 (-1)^{n-1} & & & & & \\
 p_{n-1} & (-1)^{n-1} & & & & \\
 p_{n-2} & p_{n-1} & (-1)^{n-1} & & & \\
 \vdots & & & & & \\
 p_i & & & & & \\
 & p_2 & & & & \\
 & & p_{n-1} & & & \\
 & & & & & (-1)^{n-1}
 \end{bmatrix}
 \begin{bmatrix}
 v^T u \\
 v^T A u \\
 v^T A^2 u \\
 \vdots \\
 v^T A^{n-1} u
 \end{bmatrix}
 =
 \begin{bmatrix}
 r_{n-1} \\
 r_{n-2} \\
 r_{n-3} \\
 \vdots \\
 r_0
 \end{bmatrix}$$

Soit $PX = R$ avec $X_i = v^T A^i u$. $i = 0, \dots, n-1$

P étant inversible on a : $X = P^{-1}R$ et soient $\xi_0, \xi_1, \dots, \xi_{n-1}$ les solutions du système $PX = R$.

Notre problème est alors de déterminer 2 vecteurs u et v de R^n tels que :

$$\text{(S)} \quad \begin{cases}
 v^T u = \xi_0 \\
 v^T A u = \xi_1 \\
 \vdots \\
 v^T A^{n-1} u = \xi_{n-1}
 \end{cases}$$

III - EXISTENCE D'UNE SOLUTION

Nous supposons que A est diagonalisable et admet n valeurs propres distinctes.

Soit $A = HDH^{-1}$ avec D : diagonale des valeurs propres.

Les relations (S) deviennent alors :

$$(S') \quad \begin{cases} v^T H H^{-1} u = \xi_0 \\ v^T H D H^{-1} u = \xi_1 \\ \vdots \\ v^T H D^{n-1} H^{-1} u = \xi_{n-1} \end{cases}$$

$$\text{et en posant : } \begin{cases} y = H^T v \\ x = H^{-1} u \end{cases}$$

$$(S') \text{ s'écrit : } \begin{cases} y^T x = \xi_0 \\ y^T D^i x = \xi_i \\ \vdots \\ y^T D^{n-1} x = \xi_{n-1} \end{cases}$$

soit sous forme matricielle :

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ \lambda_1 & \lambda_2 & \dots & \lambda_n \\ \lambda_1^2 & \lambda_2^2 & \dots & \lambda_n^2 \\ \vdots & \vdots & \dots & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \dots & \lambda_n^{n-1} \end{bmatrix} \begin{bmatrix} x_1 y_1 \\ x_2 y_2 \\ x_3 y_3 \\ \vdots \\ x_n y_n \end{bmatrix} = \begin{bmatrix} \xi_0 \\ \xi_1 \\ \xi_2 \\ \vdots \\ \xi_{n-1} \end{bmatrix}$$

Le déterminant de la matrice de premier membre est un déterminant de Van der Monde

et il est non nul puisque $\lambda_i \neq \lambda_j$ pour $i \neq j$. Ce système admet donc une solution soit :

$$\begin{cases} x_1 y_1 = s_1 \\ x_2 y_2 = s_2 \\ \vdots \\ x_n y_n = s_n \end{cases}$$

Le problème posé admet donc une infinité de solutions u et v et nous rechercherons celles qui minimisent la norme euclidienne de la matrice uv^T ce qui s'écrit :

$$\text{Min}_{u, v \in \mathbb{R}^n} \{ \|uv^T\|_E \mid v^T A^i u = \xi_i \quad i = 0, \dots, n-1 \}$$

IV - RECHERCHE DE LA SOLUTION DE NORME MINIMUM

Nous supposons A normale ($AA^T = A^T A$). Alors il existe une matrice Q unitaire telle que $Q^T D Q = A$, où D est la matrice diagonale des valeurs propres de A .

Les relations $v^T A^i u = \xi_i \quad i=0, \dots, n-1$

s'écrivent alors : $v^T Q^T D^i Q u = \xi_i \quad i = 0, \dots, n-1$ et avec le changement de base défini

par Q^T : si $Q u = x$

et $Q v = y$

$$y^T D^i x = \xi_i \quad i = 0, \dots, n-1.$$

Si nous recherchons alors les vecteurs u et v satisfaisant aux relations $v^T A^i u = \xi_i \quad i = 0, \dots, n-1$ et tels que $\|uv^T\|_E$ soit minimum il faut résoudre le problème :

$$\text{Min}_{u, v \in \mathbb{R}^n} \{ \|uv^T\|_E^2 \mid v^T A^i u = \xi_i \quad i = 0, \dots, n-1 \} \text{ ou ce qui est identique :}$$

$$(P) \quad \text{Min}_{u, v \in \mathbb{R}^n} \{ \|u\|_E^2, \|v\|_E^2 \mid v^T A^i u = \xi_i \quad i = 0, \dots, n-1 \}.$$

Remarquons que si u_0 , solution du problème (P) est multiplié par $\lambda \in \mathbb{R}$ alors v_0 est divisé par λ . Le problème (P) se ramène au problème suivant :

$$(P') \quad \text{Min}_{u, v \in \mathbb{R}^n} \{ \|v\|_E^2 \mid v^T A^i u = \xi_i \quad i=0, \dots, n-1 \\ \|u\|_E^2 = 1 \}$$

Si nous faisons le changement de base défini par Q^T :

$$Qu = x$$

$$Qv = y$$

Q , étant unitaire, elle conserve la norme euclidienne donc :

$$\|u\|_E = 1 \Leftrightarrow \|Q^T x\|_E = 1 \Leftrightarrow \|x\|_E = 1$$

et

$$\|v\|_E = \|Q^T y\|_E = \|y\|_E.$$

Le problème (P') s'écrit donc :

$$\begin{aligned} \text{Min}_{y \in \mathbb{R}^n} \{ \|y\|_E^2 \mid y^T D^i x = \xi_i \quad i = 0, \dots, n-1 \} \\ \|x\|_E^2 = 1 \end{aligned}$$

Les relations $y^T D^i x = \xi_i \quad i = 0, \dots, n-1$ s'écrivent sous forme matricielle :

$$\begin{vmatrix} 1 & 1 & \dots & 1 \\ \lambda_2^1 & \lambda_2^2 & & \lambda_2^n \\ 1 & 2 & & n \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & & \lambda_n^{n-1} \\ 1 & 2 & & n \end{vmatrix} \begin{vmatrix} x_1 y_1 \\ x_2 y_2 \\ \vdots \\ x_n y_n \end{vmatrix} = \begin{vmatrix} \xi_0 \\ \xi_1 \\ \vdots \\ \xi_{n-1} \end{vmatrix}$$

Système qui admet une solution si $\lambda_i \neq \lambda_j$ pour $i \neq j$.

Soit s_1, s_2, \dots, s_n la solution de ce système. Le problème de minimisation P' devient donc :

$$(P) \quad \begin{aligned} \text{Min}_{x, y \in \mathbb{R}^n} \{ \|y\|_E^2 \mid x_i \cdot y_i = s_i \quad i = 1 \dots n \} \\ \|x\|_E^2 = 1 \end{aligned}$$

Pour résoudre ce problème utilisons la méthode des multiplicateurs de Lagrange.

Soit :

$$f(x,y) = \sum_{i=1}^n y_i^2 + \mu(\sum_{i=1}^n x_i^2 - 1) + \sum_{i=1}^n \alpha_i (x_i \cdot y_i - s_i)$$

Pour x et $y \in \mathbb{R}^n$ solutions du problème (P) $f(x,y)$ passe par une valeur qui est le minimum cherché. Or $f(x,y)$ passe par un minimum lorsque :

$$\begin{cases} \frac{\partial f}{\partial x_i} = 0 & i = 1 \dots n \\ \frac{\partial f}{\partial y_j} = 0 & j = 1 \dots n. \end{cases}$$

$$\frac{\partial f}{\partial x_i} = 2\mu x_i + \alpha_i y_i = 0 \quad i = 1 \dots n$$

$$\frac{\partial f}{\partial y_j} = 2y_j + \alpha_j x_j = 0 \quad j = 1 \dots n$$

Considérons les 2 équations : $\frac{\partial f}{\partial x_i} = 0$ et $\frac{\partial f}{\partial y_i} = 0$

$$\begin{cases} 2\mu x_i + \alpha_i y_i = 0 \\ 2y_i + \alpha_i x_i = 0 \end{cases}$$

On peut les écrire sous la forme

$$\begin{pmatrix} 2\mu & \alpha_i \\ \alpha_i & 2 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

ce qui implique que la matrice du premier

membre doit être singulière d'où :

$$4\mu - \alpha_i^2 = 0 \quad \text{ce qui entraîne } \alpha_i = 2\sqrt{\mu}, \text{ ceci étant valable pour } i = 1 \dots n.$$

En utilisant le fait que $x_i \cdot y_i = s_i$ et que $2\frac{s_i}{x_i} + \alpha_i x_i = 0$, ce qui s'écrit :

$$2s_i + \alpha_i x_i^2 = 0 \text{ nous avons :}$$

$$x_i^2 = -2 \frac{s_i}{\alpha_i}. \text{ Nous prendrons donc pour } \alpha_i \text{ la détermination } +2\sqrt{\mu} \text{ si } s_i \text{ est}$$

négatif et $-2\sqrt{\mu}$ si s_i est positif.

D'où :

$$x_i^2 = \frac{|s_i|}{\sqrt{\mu}}$$

Comme $\sum_{i=1}^n x_i^2 = 1$ nous obtenons : $\sqrt{\mu} = \sum_{i=1}^n |s_i|$

D'où :

$$(1.4) \quad \left| \begin{array}{l} x_i = \left(\frac{|s_i|}{\sum_{i=1}^n |s_i|} \right)^{1/2} \quad i = 1 \dots n \\ y_i = s_i \left(\frac{|s_i|}{\sum_{i=1}^n |s_i|} \right)^{1/2} \quad i = 1 \dots n \end{array} \right.$$

Remarquons que la solution obtenue n'est pas unique : chaque x_i est déterminé au signe près et de même pour y_i .

Nous vérifions que $\|x\|_E^2 = 1$ et la valeur du minimum obtenue est alors :

$$\|y\|_E^2 = \sum_{i=1}^n s_i^2 \cdot \sum_{k=1}^n \frac{|s_k|}{|s_i|} = \sum_{k=1}^n |s_k| \cdot \sum_{i=1}^n |s_i| = \left(\sum_{i=1}^n |s_i| \right)^2$$

d'où :

$$\|y\|_E = \sum_{i=1}^n |s_i|$$

Ayant ainsi obtenu x et y il est possible de déterminer u et v par :

$$\left| \begin{array}{l} u = Q^T x \\ v = Q^T y \end{array} \right.$$

d'où le résultat :

Soit A une matrice normale ayant n valeurs propres distinctes. Soit z le vecteur des coefficients du polynôme caractéristique de A et soit \tilde{z} le vecteur calculé. Alors il existe au moins une matrice antiscaire uv^T , u et v $\in R^n$, telle que \tilde{z} soit le vecteur exact des coefficients du polynôme caractéristique de $A+uv^T$ et telle que $\|uv^T\|_E$ soit minimum, ce minimum étant alors égal à :

$$\text{Min}\|uv^T\| = \sum_{i=1}^n |s_i| \text{ les } s_i \text{ étant définis comme ci-dessus.}$$

V - APPLICATIONS

Nous avons utilisé ce résultat de la façon suivante :

Ayant construit une matrice A symétrique, dont les valeurs propres et les coefficients du polynôme caractéristique étaient connus, nous avons calculé les coefficients du polynôme caractéristique de A par la méthode de Leverrier. Ceci nous a donné le vecteur R du système $PX = R$ (§2), système triangulaire que nous avons inversé pour obtenir les quantités $\xi_0, \xi_1, \dots, \xi_{n-1}$.

Ayant obtenu les ξ_i , $i = 0, \dots, n-1$, nous avons inversé le système :

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ \lambda_1 & \lambda_2 & & \lambda_n \\ \vdots & \vdots & & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & & \lambda_n^{n-1} \end{bmatrix} \begin{bmatrix} x_1 y_1 \\ x_2 y_2 \\ \vdots \\ x_n y_n \end{bmatrix} = \begin{bmatrix} \xi_0 \\ \xi_1 \\ \vdots \\ \xi_{n-1} \end{bmatrix}$$

Soit $Mw = \xi$.

L'inversion de M s'est effectuée de la façon suivante. Considérons le système linéaire $M^T X = Y$: résoudre ce système linéaire revient à chercher les coefficients du polynôme de degré n-1, $P_{n-1}(t)$, qui pour $t = \lambda_1, \lambda_2, \dots, \lambda_n$ prend les valeurs y_1, y_2, \dots, y_n .

Donc si $L_i(t) = \frac{\prod_{j \neq i} (t-t_j)}{\prod_{j \neq i} (t_i-t_j)}$, polynôme de Lagrange tel que $L_i(t_j) = \delta_{ij}$,

le polynôme cherché s'écrit :

$$(1.5) \quad P_{n-1}(t) = \sum_{i=1}^n y_i L_i(t).$$

Chacun des coefficients du polynôme $P_{n-1}(t)$, a_0, a_1, \dots, a_{n-1} , est donc une fonction linéaire des y_i $i = 1 \dots n$:

$$a_j = \sum_{i=1}^n \alpha_{j,i} y_i.$$

Les $\alpha_{j,i}$ sont alors les éléments de $(M^T)^{-1}$ d'ou ceux de M^{-1} puisque $(M^T)^{-1} = (M^{-1})^T$.

Détermination des $\alpha_{j,i}$

Si nous appelons :

$$\sigma_1^j = \lambda_1 + \lambda_2 + \dots + \lambda_{j-1} + \lambda_{j+1} + \dots + \lambda_n$$

$$\sigma_2^j = \prod_{\substack{k \neq l \\ k \text{ et } l \neq j}} \lambda_k \lambda_l \quad \text{pour } j = 1 \dots n$$

$$\sigma_n^j = \lambda_1 \cdot \lambda_2 \cdot \dots \cdot \lambda_{j-1} \lambda_{j+1} \cdot \dots \cdot \lambda_n$$

$$D_j = (\lambda_j - \lambda_1)(\lambda_j - \lambda_2) \dots (\lambda_j - \lambda_{j-1})(\lambda_j - \lambda_{j+1}) \dots (\lambda_j - \lambda_n)$$

en développant l'expression (1.5) nous obtenons :

$$\alpha_{j,i} = \sigma_{n-j}^i / D_i \quad i = 1 \dots n-1$$

$$j = 1 \dots n$$

$$\alpha_{n,j} = 1/D_j$$

$$j = 1 \dots n.$$

Vecteur u : 0.28774818 + 00 Vecteur v : 0.69325 - 04
 0.26455460 - 01 -0.11107 - 03
 0.35286829 + 00 -0.35184 - 04
 -0.10822508 + 00 0.16138 - 03
 0.10509744 + 00 -0.927868 - 04
 0.46254086 + 00 0.28689 - 04
 0.52982114 + 00 0.13047 - 04
 0.52399656 + 00 +0.460 - 05

Coefficients exacts	Coefficients de A calculés	Coefficients de $A+uv^T$ calculés
1	1	1
-4	-4	-4
-1.8125	-1.8124988	-1.8125
19.25	19.24998	19.25
-11.046875	-11.046875	-11.046872
-13.5625	-13.562494	-13.562504
9.609375	9.6093712	9.6093813
2.25	2.2500044	2.2500044
-1.6875	-1.6874542	-1.6875305

Erreur relative : 0.16511981 - 05 0.11299100 - 05

b) Matrice (8,8) symétrique :

1.8125	2.3125	3.0625	2.0625	1.8125	1.5625	1.0625	-3.4375
.	2.8125	2.5625	1.5625	1.3125	1.0625	0.5625	-3.9375
.	.	1.3125	2.3125	2.0625	1.8125	1.3125	-3.1875
.	.	.	3.3125	1.0625	0.8125	0.3125	-4.1875
.	.	.	.	3.8125	0.5625	-0.0625	-4.4375
.	4.3125	-0.1875	-4.6875
.	5.3125	-5.1875
.	14.3125

Coefficients exacts	Coefficients calculés	Coefficients de $A+uv^T$
1	1	1
-37	-37	-37
361	361	361
-1183	-1183	-1183
-98	-98	-97.998047
6692	6692.0124	6692
-7176	-7175.3333	-7178
-5472	-5476.5714	-5494.2857
6912	7679.9999	7132.4624

Erreur relative :

0.57967498 - 01

0.29324578 - 01

c) Dans cet exemple nous nous sommes donnés une matrice A de coefficients de polynôme caractéristique donné et nous avons essayé de déterminer $A' = A+uv^T$ de telle sorte que les coefficients du polynôme caractéristique de A' soient identiques à ceux de A calculés sauf le dernier.

A symétrique (6 6)

1.2222222	1.5555555	2.5555555	1.2222222	0.5555555	-1.4444444
	1.8888888	1.8888888	0.5555556	-0.1111111	-2.1111111
		0.8888888	1.5555556	0.8888888	-1.1111111
			+2.2222222	-0.4444444	-2.4444444
				2.8888889	-3.1111111
					4.8888890

Coefficients exacts de A :

1
-14
35
70
-196
-56
160

Coefficients calculés :

1
-14
35
69.999985
-195.99997
-56.00039
160.00911

Nous cherchons A' ayant pour coefficients ceux de A calculés sauf le dernier que nous imposons à être égal à 140.

On a alors :

vecteur u :	0.28338505	vecteur v :	-0.15182368
	0.1004687		0.19243121
	0.44701210		0.77706863 - 01
	0.15031186		-0.19587795
	0.47512737		0.68399221 - 01
	0.67929338		-0.20757068 - 01

et si l'on calcule les coefficients du polynôme caractéristique de A' on trouve alors :

1
-14
35
70
-196.00058
<u>-55.994726</u>
<u>139.95443</u>

CHAPITRE - VII

METHODE DE NEWTON POUR LE PROBLEME DES ELEMENTS

PROPRES D'UNE MATRICE SYMETRIQUE

I - PRINCIPE DE LA METHODE

1 - NOTATIONS

Nous noterons :

$\mathcal{M}_{n,n}$: l'algèbre des matrices carrées à éléments réels.

$\mathcal{U}_{n,n}$: les matrices du groupe unitaire.

\mathcal{S}_n : le sous-espace des matrices symétriques de $\mathcal{M}_{n,n}$

\mathcal{D}_n : la sous algèbre des matrices diagonales de $\mathcal{M}_{n,n}$

$\mathcal{E}_{s,n}$: la sous algèbre des matrices triangulaires supérieures de $\mathcal{M}_{n,n}$

\mathcal{AS}_n : sous espace des matrices antisymétriques.

2 - EXPOSE DE LA METHODE

Soit $A \in \mathcal{S}_n$ fixée. Si $Q \in \mathcal{U}_n$, l'ensemble Γ_A des matrices $B = Q^T A Q$ est une variété de $\mathcal{M}_{n,n}$. Q_0 étant donnée, cherchons quelle est la variété linéaire affine tangente à Γ_A en son point $B_0 = Q_0^T A Q_0$.

a) Différentielle (de Frechet) de l'application : $\phi_A : Q \in \mathcal{M}_{n,n} \rightarrow Q^T A Q \in \mathcal{M}_{n,n}$

D'une part :

$$(Q_0 + X)^T A (Q_0 + X) = Q_0^T A Q_0 + X^T A Q_0 + Q_0^T A X + X^T A X.$$

D'autre part, si $\| \cdot \|$ désigne une norme de matrice quelconque sur $\mathcal{M}_{n,n}$ on a :

$$\lim_{\|X\| \rightarrow 0} \left(\frac{\|X^T A X\|}{\|X\|} \right) = 0.$$

La différentielle (au sens de Fréchet) de ϕ_A en Q_0 est donc l'application linéaire :

$$L_A(Q_0) : X \in \mathcal{M}_{N,N} \rightarrow X^T A Q_0 + Q_0^T A X \in \mathcal{M}_{n,n}$$

b) Variété affine tangente à $\mathcal{U}_{n,n}$ en Q_0

Nous avons vu ci-dessus que l'application

$$\phi_I : Q \in \mathcal{M}_{n,n} \rightarrow Q^T Q \in \mathcal{M}_{n,n} \text{ admet pour différentielle en } Q_0 :$$

$$L_I(Q_0) : X \in \mathcal{M}_{n,n} \rightarrow X^T Q_0 + Q_0^T X.$$

Donc si l'on restreint Q à appartenir à $\mathcal{U}_{n,n}$ on a $Q^T Q = I$ et par suite : $X^T Q_0 + Q_0^T X = 0$.

Donc la variété affine tangente à $\mathcal{U}_{n,n}$ en Q_0 est :

$$W_{Q_0} = \{Q_0 + X ; Q_0^T X + X^T Q_0 = 0\}.$$

c) Variété affine tangente à Γ_A au point $B_0 = Q_0^T A Q_0$

De a) et b) on déduit que la variété affine tangente à Γ_A au point $B_0 = Q_0^T A Q_0$ n'est autre que :

$$V_{Q_0} = \{Q_0^T A Q_0 + X^T A Q_0 + Q_0^T A X \text{ avec } X^T Q_0 + Q_0^T X = 0\}.$$

La méthode que nous proposons, comme celle de Newton pour résoudre de simples équations scalaires, est d'étudier l'intersection de V_{Q_0} avec \mathcal{D}_n , c'est-à-dire trouver X tel que :

$$\mathcal{P}(0) \begin{cases} Q_0^T A Q_0 + X^T A Q_0 + Q_0^T A X = D \in \mathcal{D}_n \\ X^T Q_0 + Q_0^T X = 0. \end{cases}$$

Nous prouverons que $\mathcal{P}(0)$, dans certaines conditions, admet une solution et une seule X_0 . On considèrera alors une nouvelle matrice unitaire Q_1 déterminée par :

$$\mathcal{Q}(0) \quad \{ Q_0 + X_0 = Q_1 \cdot R_1 \text{ avec } Q_1 \in \mathcal{U}_{n,n} \text{ et } R_1 \in \mathcal{C}_{s,n}.$$

L'algorithme étudié est le suivant :

Connaissant Q_n trouver X_n puis R_{n+1} , Q_{n+1} tels que :

$$\mathcal{P}(n) \begin{cases} Q_n^T A Q_n + X_n^T A Q_n + Q_n^T A X_n = D_n \in \mathcal{D}_n \\ X_n^T Q_n + Q_n^T X_n = 0 \end{cases}$$

$$\mathcal{Q}(n) \quad \{ Q_n + X_n = Q_{n+1} \cdot R_{n+1} \quad Q_{n+1} \in \mathcal{U}_{n,n}, R_{n+1} \in \mathcal{C}_{s,n}.$$

II - PROBLEME $\mathcal{P}(n)$

Pour chaque valeur de n nous avons à résoudre un problème du type :
déterminer X telle que

$$\mathcal{P} \begin{cases} Q^T A Q + X^T A Q + Q^T A X = D \in \mathcal{D}_N \\ X^T Q + Q^T X = 0. \end{cases}$$

Soit $B = Q^T A Q \in \mathcal{J}_n$ et $X^T Q = Y$. Notre problème est donc de trouver $Y \in \mathcal{M}_{n,n}$ telle que :

$$\mathcal{P} \begin{cases} B + YB + BY^T = D \in \mathcal{D}_n \\ Y + Y^T = 0. \end{cases}$$

La deuxième condition nous montre que $Y \in \mathcal{A}_{\mathcal{J}_n}$ le sous-espace des matrices anti-symétriques. Si $\{E_{ij}\}$ désigne l'ensemble des matrices de base de \mathcal{M}_{nn} $((E_{i,j})_{k,l} = 0,$

$(E_{i,j})_{i,j} = 1$) l'ensemble

$\{E_{ij} - E_{ji} ; i > j \quad j = 1 \dots n\}$ est une base pour $\mathcal{A} \mathcal{S}_n$ et Y s'écrit :

$$Y = \sum_{i>j} y_{ij} (E_{ij} - E_{ji}) \in \mathcal{A} \mathcal{S}_n \text{ est un sous-espace à } \frac{n(n-1)}{2} \text{ dimensions.}$$

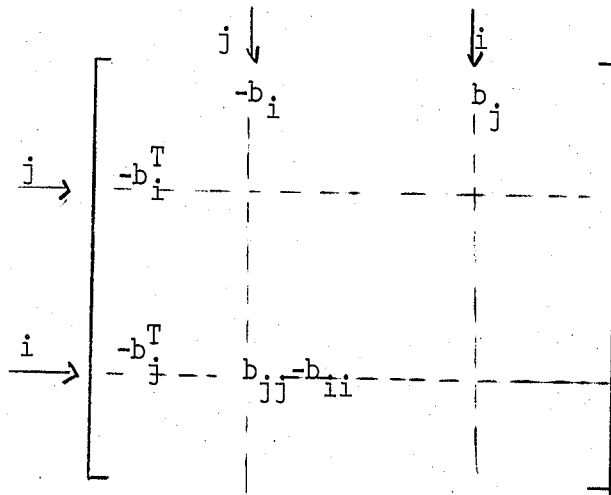
Puisque la matrice $B + YB + BY^T$ est symétrique, pour écrire qu'elle appartient à \mathcal{D}_n , il suffira d'écrire $\frac{n(n-1)}{2}$ relations par exemple en écrivant que les termes sous la diagonale de celle-ci sont nuls. Cela donnera les $\frac{n(n-1)}{2}$ équations linéaires définissant les valeurs de Y_{ij} .

Plus précisément écrivons :

$$(S) \quad B + \sum_{i>j} y_{ij} [(E_{ij} - E_{ji})B + B(E_{ji} - E_{ij})] \in \mathcal{D}_n$$

soit

$$(S) \quad B + \sum_{i>j} y_{ij} C_{ij} \in \mathcal{D}_n \quad C_{i,j} \text{ étant la matrice formée ainsi}$$



où b_k désigne la $k^{\text{ième}}$ colonne de la matrice B.

Les éléments de $C_{i,j}$ sont ceux des colonnes (ou des lignes) de rang i ou j de B placées dans la disposition ci-dessus.

Remarque importante

L'élément en position (i,j) de $C_{i,j}$ n'est autre que $b_{jj} - b_{ii}$, les autres éléments au dessous (strictement) de la diagonale de $C_{i,j}$ sont soit de la forme $-b_{li}$ ($l \neq i$) soit de la forme b_{lj} ($l \neq j$).

Supposons que l'ordination choisie pour les composantes de Y soit par exemple : $(y_{21} \ y_{31} \dots y_{N1}, y_{32} \dots y_{N2}, y_{43}, \dots, y_{n,n-1}) = Y^T$.

Le système linéaire d'ordre $\frac{n(n-1)}{2}$ donnant les $y_{i,j}$ s'écrit :

$$(1) \quad \Sigma y = u$$

où $u^T = (-b_{21}, -b_{31}, \dots, -b_{N1}, -b_{32}, \dots, -b_{N2}, -b_{43}, \dots, \dots, -b_{n,n-1})$

et où la matrice Σ de type $(\frac{n(n-1)}{2}, \frac{n(n-1)}{2})$ est telle que son élément diagonal correspondant au rang (i,j) (c'est-à-dire avec l'ordination choisie pour les y_{ij} le coefficient de la $k^{\text{ième}}$ inconnue dans le $k^{\text{ième}}$ équation avec $k = \frac{1}{2} (j-1) (2n-j) + i-j$) est $b_{jj} - b_{ii}$, les éléments hors diagonaux étant soit $-b_{li}$ ($l \neq i$) soit b_{lj} ($l \neq j$).

Supposons alors que la condition suivante ait lieu pour la matrice B :

$$\mathcal{C} : \forall i \neq j \quad |b_{ii} - b_{jj}| > \sum_{l \neq i} |b_{li}| + \sum_{l \neq j} |b_{lj}| = \sum_{l \neq i} |b_{il}| + \sum_{l \neq j} |b_{jl}|$$

Proposition 1

Si la condition \mathcal{C} est réalisée pour la matrice B , le système $\Sigma y = u$ (donc \mathcal{P}) admet une et une seule solution telle que

$$(2) \quad \text{Max}_{i>j} |y_{ij}| < \frac{m}{1-\mu} \text{Max}_{i,j} |b_{i,j}|, \quad \mu \text{ et } m \text{ étant définis ci-après}$$

par (4) et (5).

Démonstration

Soit $\Sigma = \Delta + \Phi$, Δ étant la matrice diagonale formée des éléments diagonaux de Σ et Φ de la partie "hors diagonale" de Σ . \mathcal{C} implique tout d'abord que Δ^{-1} existe ; donc :

$$(3) \quad \Sigma = \Delta(I + \Delta^{-1}\Phi).$$

D'autre part, si $\varphi(\xi) = \max_{k=1, \dots, \frac{n(n-1)}{2}} (|\xi_k|)$ est la norme sur $\mathbb{R}^{\frac{n(n-1)}{2}}$ du maximum de la valeur absolue des composantes d'un vecteur $\xi \in \mathbb{R}^{\frac{n(n-1)}{2}}$, et si

$S_{\varphi\varphi}(A) = \max_{\xi} \left(\frac{\varphi(A\xi)}{\varphi(\xi)} \right)$ est la norme de matrice associée on sait que :

$$S_{\varphi\varphi}(M) = \max_{k=1, \dots, \frac{n(n-1)}{2}} \left(\sum_{r=1}^{n(n-1)/2} |m_{kr}| \right).$$

Donc d'après \mathcal{C} on a :

$$(4) \quad \mu = S_{\varphi\varphi}(\Delta^{-1}\Phi) = \max_{i>j} \frac{(\sum_{l \neq i} |b_{il}| + \sum_{l \neq j} |b_{jl}|)}{|b_{ii} - b_{jj}|} < 1.$$

Cela prouve que $\Delta^{-1}\Phi$ est contractante dans $\mathbb{R}^{\frac{n(n-1)}{2}}$ et puis que :

$$(I + \Delta^{-1}\Phi)^{-1} = I - \Delta^{-1}\Phi + (\Delta^{-1}\Phi)^2 - \dots + (-1)^n (\Delta^{-1}\Phi)^n + \dots$$

$$\text{Donc } S_{\varphi\varphi}((I + \Delta^{-1}\Phi)^{-1}) \leq 1 + \mu + \mu^2 + \dots + \mu^n + \dots = \frac{1}{1-\mu} = \frac{1}{1 - S_{\varphi\varphi}(\Delta^{-1}\Phi)}$$

Donc Σ^{-1} existe et c'est $(I + \Delta^{-1}\Phi)^{-1} \Delta^{-1}$. De plus

$$S_{\varphi\varphi}(\Sigma^{-1}) \leq \frac{S_{\varphi\varphi}(\Delta^{-1})}{1 - S_{\varphi\varphi}(\Delta^{-1}\Phi)} = \frac{m}{1-\mu} \text{ si l'on pose}$$

$$(5) \quad m = \max_{i>j} \left(\frac{1}{|b_{ii} - b_{jj}|} \right).$$

Il suffit alors de remarquer que $y = \Sigma^{-1}u$ implique :

$$(y) \leq S_{\varphi}(\Sigma^{-1}).\varphi(u) \text{ pour obtenir (2).}$$

III - PROBLEME DE TYPE $\mathcal{G}(n)$

Q étant donnée, on forme $B = Q^T A Q$ puis on détermine Y par (C) puis on forme $X = QY^T$. Enfin R' et Q' sont déterminés par :

$$Q+X = Q'.R' = Q(I+Y^T) \quad Q' \in \mathcal{M}_{n,n} \quad R' \in \mathcal{C}_{s,n}.$$

On cherche ensuite $B' = Q'^T A Q'$.

Or on a :

$$\begin{cases} Q^T A Q + X^T A Q + Q^T A X = D \in \mathcal{D}_n \\ X^T Q + Q^T X = 0 \end{cases}$$

c'est-à-dire :

$$(Q+X)^T A (Q+X) - X^T A X = D$$

$$(Q+X)^T (Q+X) - X^T X = I$$

ou

$$\begin{cases} R'^T Q'^T A Q' R' - Y Q^T A Q Y^T = D \\ R'^T R' - Y Y^T = I. \end{cases}$$

Posons $R' = I-Z$. D'après la 2ème relation ci-dessus :

$$(I-Z)^T (I-Z) = Y Y^T + I \text{ donc}$$

$$(1) \quad Z^T Z - Z^T - Z = Y Y^T$$

Soit γ une norme sous-multiplicative de $\mathcal{M}_{n,n}$ et soit Ψ l'application

$$\Psi : Z \in \mathcal{C}_{s,n} \rightarrow Z+Z^T \in \mathcal{J}_n.$$

C'est un isomorphisme d'espaces vectoriels de $\mathcal{C}_{s,N}$ sur \mathcal{J}_n . Donc il existe 2 constantes α et $\beta > 0$ telles que :

$$\forall Z \in \mathcal{C}_{s,n} : \alpha\gamma(Z) \leq \gamma(Z+Z^T) \leq \beta\gamma(Z). \quad (a)$$

D'après (1) on a :

$$\gamma(Z+Z^T) - \gamma(Z^T Z) \leq \gamma(Y) \cdot \gamma(Y^T) \quad (b).$$

Supposons de plus γ telle que $\gamma(M) = \gamma(M^T)$ (réalisé pour la norme associée à la norme euclidienne sur \mathbb{R}^n par exemple).

Alors $\gamma(Z^T \cdot Z) \leq \gamma^2(Z)$. Donc dès que :

$$(3) \quad \gamma(Z) \leq \alpha' < \alpha \quad \text{on a :}$$

$$\gamma^2(Z) \leq \alpha' \gamma(Z) \text{ et puisque } \alpha\gamma(Z) \leq \gamma(Z+Z^T)$$

$$(\alpha - \alpha')\gamma(Z) \leq \gamma(Z+Z^T) - \gamma^2(Z) \leq \gamma(Z+Z^T) - \gamma(Z^T Z) \leq \gamma^2(Y).$$

Par suite $R' = I - Z$ sera tel que $\gamma(Z) \leq \frac{\gamma^2(Y)}{\alpha - \alpha'}$ du moins dès que $\gamma(Z) \leq \alpha'$.

Si Y est tel que $\gamma^2(Y) \leq \alpha'(\alpha - \alpha')$ et si on prend $\alpha' = \frac{\alpha}{2}$ on voit que cela est vrai si $\gamma(Y) \leq \frac{\alpha}{2}$.

Conclusion :

$$\text{Si } \gamma(Y) \leq \frac{\alpha}{2} \text{ et } \gamma(Z) \leq \frac{\alpha}{2} \text{ on aura : } \gamma(Z) \leq \frac{2}{\alpha} \gamma^2(Y).$$

Cela posé soit $R'^T B' R' - Y B Y^T = D = B + Y B + B Y^T$

$$B' = (R'^T)^{-1} (B + Y B + B Y^T + Y B Y^T) (R')^{-1}$$

$$= (I - Z^T)^{-1} (I + Y) B (I + Y^T) (I - Z)^{-1}$$

$$B' = (I - Z^T)^{-1} (D + Y B Y^T) (I - Z)^{-1}.$$

Dès que $\gamma(Z)$ (c'est-à-dire $\gamma(Y)$) est assez petite on voit que :

$$B' = (I+Z^T+\dots) (D+YBY^T) (I+Z+\dots) = D+Z^T D + DZ + YBY^T+\dots$$

d'où

$$B' = B+YB + BY^T + Z^T D + DZ + YBY^T+\dots$$

$$B' = D + Z^T(B+YB+BY^T) + (B+YB+BY^T)Z+\dots$$

Soit U' la matrice composée des éléments hors diagonaux de B' et ayant des zéros sur sa diagonale. D étant diagonale on voit que :

$$\gamma(U') \leq k_1 \gamma(Z) \cdot \gamma(B) \quad k_1 \text{ étant une constante.}$$

Puisque $B = Q^T A Q$ $\gamma(B) \leq k_2 \gamma(A)$, k_2 étant une autre constante.

En conclusion :

$$\gamma(U') \leq k_1 \cdot k_2 \gamma(A) \cdot \gamma(Z).$$

Or si l'on considère u' le vecteur formé des éléments b'_{ij} (dans l'ordre indiqué plus haut), il est clair que l'isomorphisme $u' \rightarrow U'$ est tel qu'il existe des constantes $\bar{\alpha}$ et $\bar{\beta}$ telles que :

$$\bar{\alpha} \varphi(u') \leq \gamma(U') \leq \bar{\beta} \varphi(u') \text{ et de même des constantes } \bar{\alpha} \text{ et } \bar{\beta}$$

telles que :

$$\bar{\alpha} \varphi(y) \leq \gamma(Y) \leq \bar{\beta} \varphi(y).$$

$$\text{Donc } \gamma(U') \leq k_1 k_2 \gamma(A) \frac{2}{\alpha} \gamma^2(Y) \leq 2 \frac{k_1 k_2}{\alpha} \gamma(A) \bar{\beta}^2 \gamma^2(y)$$

$$\leq \frac{2k_1 k_2}{\alpha} \gamma(A) \bar{\beta}^2 \frac{m^2}{(1-\mu)^2} \varphi^2(u) \leq \frac{2k_1 k_2}{\alpha} \bar{\beta}^2 \gamma(A) \frac{m^2}{(1-\mu)^2} \frac{1}{\bar{\alpha}^2} \gamma^2(U)$$

ce qui s'écrit $\gamma(U') \leq \frac{K m^2}{(1-\mu)^2} \gamma(U) \cdot \gamma(U)$. Donc si :

$$(4) \quad \gamma(U) \frac{K m^2}{(1-\mu)^2} \leq \rho < 1 \text{ on aura certainement } \gamma'(U') \leq \rho \cdot \gamma(U).$$

D'après la définition de μ et m on voit que si $\gamma(U)$ est suffisamment petit, $\frac{m^2}{(1-\mu)^2}$ est voisin de m^2 donc dès que $\gamma(U)$ est suffisamment petit, (4) est certainement satisfait.

IV - CONVERGENCE DE PROCÉDE

Théorème

Pour Q_0 tel que $B_0 = Q_0^T A Q_0$ ait des éléments non diagonaux suffisamment petit, le procédé décrit au paragraphe I converge et la limite de B_n n'est autre que la diagonale Δ des valeurs propres de A (supposées distinctes).

1. Toutes les matrices B_n sont de la forme $Q_n^T A Q_n$ donc $\gamma(B_n) < K \cdot \gamma(A)$: les matrices B_n sont dans une bande de n, n . Cette bande étant compacte, il existe une sous-suite B_{n_i} ($i \rightarrow \infty$) qui converge vers B^* . B^* est diagonale. En effet soit $B_0 = U_0 + D_0$ U_0 étant formée des éléments hors diagonaux de B_0 : il résulte du paragraphe 3 que si $U' = Y_1$ ($U = U_0$ étant tel que $\gamma(U_0)$ soit assez petit) est tel que $\gamma(U_1) < \rho \gamma(U_0)$, et à fortiori $\gamma(U_{n_i}) < \rho^{n_i} \gamma(U_0)$, il suffit de faire tendre i vers l'infini pour voir que la partie hors diagonale de B^* doit être nulle.

2. Toute sous-suite convergente, extraite de la suite B_n , ne peut donc converger que vers B^* . Il en résulte que B_n converge vers B^* car sinon pour un $a \neq 0$, il existerait une sous-suite infinie de $\{B_n\}$ telle que $\gamma(B^* - B_n) > a$; de celle-ci on extrairait une sous-suite convergente B_{n_i} vers B^* (forcément) ce qui est absurde.

V - RESOLUTION NUMERIQUE DES PROBLEMES $\mathcal{P}(n)$ ET $\mathcal{Q}(n)$

1 - PROBLEME $\mathcal{P}(n)$:

Ce problème se ramène à la résolution du système linéaire $\Sigma y = u$ où Σ est à diagonale prépondérante (condition \mathcal{C} du paragraphe II), tout au moins lorsque $B_0 = Q_0^T A Q_0$ (avec $Q_0 = I$) soit $B_0 = A$ satisfait à la condition .

Dans ce cas nous résoudrons le système $\Sigma y = u$ par une relaxation de Gauss-Seidel, étant assurés de la convergence de l'itération, d'après le théorème de Geiringer : pour que la relaxation de Gauss-Seidel converge il suffit que toutes les quantités $\sum_{e \neq i} \frac{|\xi_{ie}|}{|\xi_{ii}|}$ soient inférieures à 1 pour $i = 1 \dots n$. (ξ_{ij} élément en position i, j de Σ). [10].

Cette mesure offre de plus l'avantage de ne pas avoir à construire explicitement la matrice Σ et de n'utiliser que 3 tableaux (n, n) : le tableau B, le tableau Y et un tableau pour les résidus R.

Soit Y le tableau :

		⋮		⋮		
		k		l		
	0					
	y ₂₁	0				
	y ₃₁	y ₃₂	0			
	y ₄₁	y ₄₂	y ₄₃	0		
	⋮					
k	y _{k1}	y _{k2}	y _{k3}	⋯	y _{k,k-1}	
	⋮				y _{k+1,k}	
l	y _{l1}	y _{l2}	y _{l3}	⋯	y _{l,k-1}	y _{l,k}
	⋮				y _{l,l-1}	0
	y _{n1}	y _{n2}	y _{n3}	⋯	y _{n,k}	y _{n,l-1}
					y _{n,l}	y _{n,l+1,l}
					⋮	y _{n,l}
					y _{n,k}	y _{n,l-1}
						y _{n,l}
						y _{n,n-1}
						0

Avec l'ordination choisie au paragraphe II la relaxation se fera sur les $y_{i,j}$ dans l'ordre $(y_{21}, y_{31}, \dots, y_{41}, y_{32}, \dots, y_{nr}, \dots, y_{n,n-1})$. Considérons alors la relaxation sur l'élément $y_{\ell,k}$ ($\ell > k$).

Si $p = \frac{1}{2} (k-1) (2n-k) + \ell - k$, cette relaxation correspond à celle sur la $p^{\text{ième}}$ inconnue de la $p^{\text{ème}}$ équation du système $\Sigma y = u$.

Quels sont alors les termes $y_{i,j}$ figurant dans cette $p^{\text{ième}}$ équation et leurs coefficients ?

Ce sont tous les $y_{i,j}$ pour lesquels le terme en position (ℓ, k) de la matrice $C_{i,j}$ correspondante n'est pas nul, c'est-à-dire :

- a) $y_{i,\ell}$ $i = \ell+1, \dots, n$ le coefficient étant alors $-b_{k,i}$
- b) $y_{\ell,j}$ $j = 1, \dots, k-1, k+1, \dots, \ell-1$ avec le coefficient : $b_{k,j}$
- c) $y_{i,k}$ $i = k+1, \dots, \ell-1, \ell+1, \dots, n$ avec le coefficient : $-b_{\ell,i}$
- d) $y_{k,j}$ $j = 1, \dots, k-1$ avec le coefficient : $b_{\ell,j}$.

La relaxation sur $y_{\ell,k}$ fait donc intervenir les éléments des lignes et des colonnes de la matrice Y matérialisées ci-dessus. De cette façon nous pouvons résoudre le système $\Sigma y = u$ par la relaxation sans avoir à construire explicitement le tableau Σ d'ou un gain de place appréciable, les résidus étant alors conserver dans un tableau R de même type que le tableau Y .

Le lecteur trouve à la fin du chapitre une procédure ALGOL correspondant à la résolution du système $\Sigma y = u$.

2 - PROBLEME (n)

Ayant déterminé Y de la façon ci-dessus, il nous faut alors décomposer la matrice : $Q+X = Q(I+Y^T)$ sous la forme $Q'.R'$ $Q' \in \mathcal{U}_{n,n}$ et $R' \in \mathcal{C}_{s,n}$.

Cette décomposition est toujours possible pour une matrice $M \in \mathcal{M}_{n,n}$ non singulière [11]. On démontre en effet que :

Théorème :

Pour toute matrice $M \in \mathcal{M}_{n,n}$ non singulière, il existe une décomposition $M = M'.T$ où T est triangulaire supérieure et unitaire, M' orthogonale en colonnes.

et de là on déduit [12] le théorème :

Pour toute matrice $M \in \mathcal{M}_{n,n}$, non singulière il existe une décomposition de la forme $M = Q.R$ avec R triangulaire supérieure et Q unitaire.

La matrice Q s'obtient à partir de la matrice M' de la décomposition de M en $M = M'.T$ (M' orthogonale en colonnes) en divisant chaque colonne de M' par sa norme $\|M'.i\|$.

Ainsi :

$$Q.i = \frac{M'.i}{\|M'.i\|} \quad i = 1 \dots n.$$

Nous utilisons ce résultat pour décomposer la matrice $Q+X$ sous la forme $Q'.R'$: à l'aide du procédé d'orthogonalisation de Schmidt nous décomposons $Q+X$ en $U'.T$, U' orthogonale en colonnes et T triangulaire supérieure puis nous déterminons Q' par :

$$Q'.i = \frac{U'.i}{\|U'.i\|} \quad i = 1 \dots n.$$

Une procédure ALGOL assurant cette décomposition est donnée à la fin de ce chapitre.

3 - ALGORITHME COMPLET

L'algorithme complet recherchant la forme diagonale d'une matrice symétrique à éléments réels sera le suivant :

Soit $B_0 = A(=I^T A I)$ soit $Q_0 = I$ le point de départ de l'itération.

- a) Nous recherchons alors Y telle que $B + YB + BY^T = 0$ (Gauss-Seidel).
- b) Nous formons $Q'_0 = Q_0(I + Y^T)$
- c) Nous décomposons Q'_0 sous la forme : $Q'_0 = Q_1.R_1$ (orthogonalisation de Schmidt).
- d) Nous calculons $B = Q_1^T A Q_1$.
- d) Si la norme des éléments non diagonaux de B est inférieure à la précision désirée alors nous arrêtons le processus sinon nous faisons $Q_0 = Q_1$ et nous retournons en b).

VI - PROCEDURES ALGOL

Nous donnons ici le programme complet correspondant à l'algorithme décrit au paragraphe V-3, la procédure principale étant

NEWTONMAT (A,Q,B,EPS,ETA,N) ; où :

- A : matrice à diagonaliser (symétrique)
 Q : matrice unitaire telle que $Q^T A Q = D \in \mathcal{D}_n$
 B : matrice $Q^T A Q$
 EPS : précision voulue sur la norme des éléments non diagonaux de B.
 ETA : précision désirée pour la résolution par relaxation du système $\Sigma y = u$.
 N : ordre de la matrice A ;

VII - RESULTATS ET CONCLUSION

Pour tester cette méthode nous avons construit un certain nombre d'exemples numériques que nous avons traité sur l'I.B.M. 360-65 du Laboratoire de Mathématiques Appliquées de Grenoble.

Exemple 1

Soit la matrice symétrique d'ordre 6 :

$$A = \begin{vmatrix} 3.04 & 0 & 0.28 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 \\ 0.28 & 0 & 4.96 & 0 & 0 & 0 \\ 0 & 0 & 0 & -5.55 & 0.4 & 0 \\ 0 & 0 & 0 & 0.4 & -4 & 0.4 \\ 0 & 0 & 0 & 0 & 0.4 & -2.45 \end{vmatrix}$$

En notant $A^{(k)}$ le $k^{\text{ième}}$ itéré, soit $Q_k^T A Q_k$, nous obtenons

$$A^{(8)} = \begin{bmatrix} +3.000000'+00 & -6.383714' -08 & +5.960226' -08 \\ -6.383708' -08 & +4.000000'+00 & -1.372652' -07 \\ -5.960691' -08 & -1.372653' -07 & +5.000000'+00 \\ +3.441921' -06 & +3.319274' -06 & +2.441247' -06 \\ +3.467045' -06 & +3.342431' -06 & +3.853493' -06 \\ +8.237917' -06 & +4.167514' -06 & +2.055020' -06 \\ \\ +3.441921' -06 & +3.467045' -06 & +8.237917' -06 \\ +3.319274' -06 & +3.342431' -06 & +4.167514' -06 \\ +2.441247' -06 & +3.853493' -06 & +2.055022' -06 \\ -5.649999'+00 & +1.307576' -06 & -5.125998' -06 \\ +2.123414' -07 & -4.000000'+00 & -4.768371' -07 \\ -5.424020' -06 & -3.576278' -07 & -2.349998'+00 \end{bmatrix}$$

La matrice Q_8 telle que $Q_8^T A Q_8 = A^{(8)}$ étant alors :

$$Q_8 = \begin{bmatrix} +9.899497'-01 & +4.377434'-08 & +1.414213'-01 \\ -6.382900'-08 & +1.000000'+00 & +1.372617'-07 \\ -1.414213'-01 & -1.449091'-07 & +9.899497'-01 \\ -5.525852'-07 & -4.547178'-07 & -3.345497'-07 \\ -7.420960'-07 & -4.682003'-07 & -4.144277'-07 \\ -1.385123'-06 & -5.455506'-07 & -1.742700'-07 \\ \\ +4.263276'-07 & +5.508666'-07 & +1.563863'-06 \\ +3.439663'-07 & +4.178044'-07 & +6.563018'-07 \\ +1.706483'-07 & +3.538180'-07 & +5.902431'-08 \\ +9.696971'-01 & +2.424239'-01 & +3.030459'-02 \\ -2.424243'-01 & +9.393942'-01 & +2.424240'-01 \\ +3.030144'-02 & -2.424244'-01 & +9.696971'-01 \end{bmatrix}$$

D'ou pour les valeurs propres :

Valeurs propres exactes	Valeurs propres calculées
3.000000+00	3.000000+00
4.000000+00	4.000000+00
5.000000+00	5.000000+00
-5.650000+00	-5.649999+00
-4.000000+00	-4.000000+00
-2.350000+00	-2.349998+00

Exemple 2

Soit la matrice symétrique d'ordre 8 :

$$A = \begin{bmatrix} 2.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 15.0 & 2.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.0 & 11.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -6.0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -10.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0 & 24.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 18.0 \end{bmatrix}$$

Après 10 itérations nous obtenons :

$$A^{(10)} = \begin{bmatrix} +2.000000'+00 & -6.192384' -08 & -1.536352' -06 & -3.353322' -06 \\ -6.192384' -08 & +4.000000'+00 & -1.974724' -06 & -1.704511' -06 \\ -1.536351' -06 & -1.974723' -06 & +1.582844'+01 & +4.768405' -06 \\ -3.353322' -06 & -1.704511' -06 & +3.814731' -06 & +1.017158'+01 \\ +3.926572' -06 & +3.531806' -06 & +4.771994' -06 & +3.318671' -06 \\ +6.923176' -06 & +3.240473' -06 & +4.393901' -06 & +4.180632' -06 \\ -4.529601' -05 & -5.478386' -05 & -3.849551' -06 & -4.629101' -06 \\ -5.255214' -05 & -6.794148' -05 & -6.440007' -06 & -4.836416' -06 \\ +3.926574' -06 & +6.923176' -06 & -4.529601' -05 & -5.255214' -05 \\ +3.531806' -06 & +3.240473' -06 & -5.478382' -05 & -6.794148' -05 \\ +4.771994' -06 & +4.393901' -06 & -3.849548' -06 & -6.440036' -06 \\ +3.318674' -06 & +4.180629' -06 & -4.629096' -06 & -4.836408' -06 \\ -6.033297'+00 & +3.107725' -07 & -5.483618' -06 & -7.478894' -06 \\ +3.107725' -07 & -1.000000'+01 & -6.389890' -06 & -7.514428' -06 \\ -6.139270' -06 & -6.389892' -06 & +2.403327'+01 & +7.516572' -06 \\ -7.478894' -06 & -7.514428' -06 & +7.516572' -06 & +1.800000'+01 \end{bmatrix}$$

avec Q_{10} telle que $Q_{10}^T A Q_{10} = A^{(10)}$:

$$Q_{10} = \begin{bmatrix} +1.000000'+00 & +3.112852' -08 & +1.111027' -07 & +4.103675' -07 \\ -3.115052' -08 & +1.000000'+00 & +1.669506' -07 & +2.761932' -07 \\ +5.438606' -08 & -4.856106' -08 & +9.238801' -01 & -3.826829' -01 \\ -4.216540' -07 & -3.190682' -07 & +3.826829' -01 & +9.238801' -01 \\ -5.569268' -07 & -4.428141' -07 & -2.337731' -07 & -2.157946' -07 \\ -5.769296' -07 & -2.314605' -07 & -1.701188' -07 & -2.072535' -07 \\ -2.038399' -06 & -2.721416' -06 & -4.616551' -07 & -3.269483' -07 \\ -3.284506' -06 & -4.852956' -06 & -2.965600' -06 & -6.177990' -07 \\ +4.887834' -07 & +5.769285' -07 & +2.055802' -06 & +3.284511' -06 \\ +3.520039' -07 & +2.314590' -07 & +2.734641' -06 & +4.852963' -06 \\ +1.232929' -07 & +7.785631' -08 & +3.056661' -07 & +2.503441' -06 \\ +2.727372' -07 & +2.565782'+07 & +4.880714' -07 & +1.705655' -06 \\ +9.994464' -01 & +7.205136' -08 & +3.327815' -02 & +3.524754' -07 \\ -7.834340' -08 & +1.000000'+00 & +1.877529' -07 & +2.683708' -07 \\ -3.327815' -02 & -1.902583' -07 & +9.994464' -01 & +1.234856' -06 \\ -3.111904' -07 & -2.683741' -07 & -1.245923' -06 & +1.000000'+00 \end{bmatrix}$$

Valeurs propres exactes	Valeurs propres calculées
2.000000+00	2.000000+00
4.000000+00	4.000000+00
1.582843+01	1.5828 <u>44</u> +01
1.017157+01	1.0171 <u>58</u> +01
-6.033300+00	-6.033 <u>297</u> +00
2.403329+01	2.402237+01
-1.000000+01	-1.000000+01
1.800000+01	1.800000+01

Les chiffres inexacts sont soulignés.

Exemple 3

Soit A symétrique d'ordre 8 :

A =	-12.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0
	x	-4.0	1.0	0.0	0.0	0.0	0.0	0.0
	x	x	4.0	0.0	0.0	0.0	0.0	0.0
	x	x	x	12.0	0.0	0.0	0.0	0.0
	x	x	x	x	-1.1568	0.5376	0.096	-0.028
	x	x	x	x	x	-2.8432	0.028	0.096
	x	x	x	x	x	x	1.0	0.0
	x	x	x	x	x	x	x	3.0

Nous obtenons :

-1.204161'-01	-2.094264'-06	-2.724042'-06	-3.635866'-06
-2.094266'-06	-4.123105'+00	-1.609312'-06	-2.413894'-06
-2.724042'-06	-1.609312'-06	+4.123105'+00	-3.823786'-06
-3.635866'-06	-2.413894'-06	-3.823786'-06	+1.204161'+01
-2.883334'-06	-2.318070'-06	+4.986994'-06	+4.824870'-06
-2.993811'-06	-2.611254'-07	+4.242172'-06	+3.133640'-06
-1.547042'-05	-4.970798'-06	+4.751577'-06	+5.790544'-06
-7.490082'-05	-4.496274'-06	+5.606542'-06	+8.425810'-06

$A^{(13)} =$

-2.883332'-06	-2.993808'-06	-1.547042'-05	-7.490082'-05
-2.318069'-06	-2.611257'-07	-4.970800'-06	-4.496274'-06
+4.986994'-06	+4.242175'-06	+4.751577'-06	+5.606525'-06
+4.824870'-06	+3.133642'-06	+5.790546'-06	+8.425810'-06
-1.004986'+00	+4.589126'-07	-4.395828'-07	-6.754411'-06
+1.964826'-07	-3.001666'+00	-5.026832'-07	-6.161628'-06
-4.023298'-07	-5.204928'-07	+1.004986'+00	-5.185678'-07
-6.744304'-06	-6.165354'-06	-5.176771'-07	+3.001665'+00

avec Q_{13} telle que $Q_{13}^T A Q_{13} = A^{(13)}$

+9.991368'-01	+2.704591'-07	+1.884385'-07	-4.155852'-02
-2.830979'-07	+9.925077'-01	+1.221835'-01	+2.072120'-07
-1.349645'-07	-1.221835'-01	+9.925077'-01	+4.610224'-07
+4.155852'-02	-1.382110'-07	-4.754714'-07	+9.991368'-01
-1.912622'-07	-6.910206'-07	-8.152882'-07	-3.255955'-07
-1.171298'-06	-9.310896'-07	-1.473526'-06	-5.055949'-07
-4.972819'-06	-6.271066'-07	-4.988842'-06	-9.284686'-07

$Q_{13} =$

+2.456600'-07	+3.222366'-07	+1.162956'-06	+4.935991'-06
+8.566722'-07	+3.038557'-07	+1.148255'-06	+1.237196'-06
+8.743600'-07	+5.624969'-07	+1.394008'-06	+4.884895'-06
+3.803559'-07	+2.218916'-07	+5.734934'-07	+1.138182'-06
+9.588084'-01	-2.799611'-01	+4.782133'-02	-4.663381'-03
+2.796522'-01	+9.598671'-01	+1.394801'-02	+1.599479'-02
-4.981392'-02	-1.387736'-07	+9.987588'-01	+1.751043'-07
-1.681288'-06	-1.666076'-02	-2.615509'-07	+9.998612'-01

et l'on a pour les valeurs propres :

Valeurs propres exactes	Valeurs propres calculées
-1.204160+01	-1.20416 <u>1</u> +01
-4.123106+00	-4.12310 <u>5</u> +00
+4.123106+00	4.12310 <u>5</u> +00
+1.204160+01	1.20416 <u>1</u> +01
1.004990+00	-1.0049 <u>86</u> +00
-1.004990+00	+1.0049 <u>86</u> +00
3.001660+00	3.0016 <u>65</u> +00
-3.001660+00	-3.0016 <u>65</u> +00

les chiffres inexacts étant soulignés.

CONCLUSION

Sur les exemples traités nous pouvons constater que les valeurs propres des matrices considérées sont déterminées avec une bonne précision. Nous constatons aussi que cette méthode nous fournit les vecteurs propres de la matrice traitée : ce sont les colonnes de la matrice Q_k telle que

$$Q_k^T A Q_k = A^{(k)}.$$

En dépit de la condition \mathcal{C} :

$$\mathcal{C} : \quad \forall i \neq j \quad |b_{ii} - b_{jj}| > \sum_{\ell \neq i} |b_{\ell i}| + \sum_{\ell \neq j} |b_{\ell j}| = \sum_{\ell \neq i} |b_{i\ell}| + \sum_{j \neq \ell} |b_{j\ell}|$$

qui restreint le champ des applications de cette méthode à des matrices ayant toutes leurs valeurs propres distinctes, nous pensons qu'elle peut être un moyen pratique de déterminer les valeurs propres et les vecteurs propres d'une matrice symétrique.

Ces limitations sont d'ailleurs les mêmes que, pour la recherche de la solution de $f(x) = 0$ par la méthode de Newton, qui se trouve en défaut dans le cas d'une racine double, ou de racines assez voisines.

BIBLIOGRAPHIE

- [1] N. GASTINEL - Matrice du second degré et normes générales en analyse numérique linéaire.
Publications Scientifiques et Techniques du Ministère de l'Air. Paris N.T. 110 - S.D.I.T. (1962) p. 5.
- [2] N. GASTINEL - Analyse Numérique Linéaire
Hermann - Paris 1966 - p. 218.
- [3] M. DUC-JACQUET - Résolution de quelques problèmes d'analyse numérique linéaire à l'aide de perturbations par des matrices antisymétriques.
Thèse de 3ème Cycle - Université de Grenoble - Janvier 1968.
p. 71
- [4] E. DURAND - Solutions numériques des équations algébriques.
Masson 1960, Tome I.
- [5] J.A. SERRET - Cours d'algèbre supérieure
Gauthier Villars 1877 - Tome I.
- [6] J. Mac CARTHY - LISP 1-5 Programmer's Manual
MIT Press Cambridge Mass. 02139.
- [7] BERKEKEY and BOBROW - The programming language LISP, tis applications and operations.
Information Intercontinental, Inc. Cambridge Mass. 02143.
- [8] C. WEISSMAN - LISP 1-5. Primer.
Dickenson publishing Company. Inc. Belmont California.
- [9] N. GASTINEL - Op. Cit 2 p.
- [10] N. GASTINEL - Op. Cit 2 p.
- [11] N. GASTINEL - Op. Cit 2 p. 95
- [12] N. GASTINEL - Op. Cit 2 p. 98

VI

Grenoble, le

Le Président de la Thèse

VU

Grenoble, le

Le Doyen de la Faculté des Sciences

Vu, et permis d'imprimer,

Le Recteur de l'Académie de GRENOBLE

'DEBUT'

'PROCEDURE' IMPMAT(A,M,N) FF 'VALEUR' M,N FF

'REEL' 'TABLEAU' A FF

'ENTIER' M,N FF

'DEBUT' 'ENTIER' I,J,K,L FF

'AIGUILLAGE' AIG F= DO,RE,MI,FA,SOL,LA,SI,UT FF

'SI' N 'SUPER' 8 'ALORS' 'ALLERA' ZUT FF

'POUR' I F= 8 'PAS' -1 'JUSQUA' 1 'FAIRE'

'SI' ABS(N-1) = 0 'ALORS'

'DEBUT'

ECRIRE(,,, 'MATRICE A M LIG',

'NES ET N COLONN', 'ES') FF

ECRIRE(,,, '*****', '*****') FF

SAUTLIGNE FF

ECRIRE(,,, M = 'M',

N = 'N',

'N) FF SAUTLIGNE FF

'ALLERA' AIG.(I). FF

'FIN' FF

DOF 'POUR' J F= 1 'PAS' 1 'JUSQUA' M 'FAIRE'

ECRIRE(,,, A.(J,1).) FF 'ALLERA' END FF

REF 'POUR' J F= 1 'PAS' 1 'JUSQUA' M 'FAIRE'

ECRIRE(,,, A.(J,1), A.(J,2).) FF 'ALLERA' END FF

MIF 'POUR' J F= 1 'PAS' 1 'JUSQUA' M 'FAIRE'

ECRIRE(,,, A.(J,1), A.(J,2), A.(J,3).) FF 'ALLERA' END FF

FAF 'POUR' J F= 1 'PAS' 1 'JUSQUA' M 'FAIRE'

ECRIRE(,,, A.(J,1), A.(J,2), A.(J,3), A.(J,4).) FF

'ALLERA' END FF

SOLF 'POUR' J F= 1 'PAS' 1 'JUSQUA' M 'FAIRE'

ECRIRE(,,, A.(J,1), A.(J,2), A.(J,3), A.(J,4), A.(J,5).) FF

'ALLERA' END FF

LAF 'POUR' J F= 1 'PAS' 1 'JUSQUA' M 'FAIRE'

ECRIRE(,,, A.(J,1), A.(J,2), A.(J,3), A.(J,4), A.(J,5),

A.(J,6).) FF 'ALLERA' END FF

SIF 'POUR' J F= 1 'PAS' 1 'JUSQUA' M 'FAIRE'

ECRIRE(,,, A.(J,1), A.(J,2), A.(J,3), A.(J,4), A.(J,5),

A.(J,6), A.(J,7).) FF 'ALLERA' END FF

UTF 'POUR' J F= 1 'PAS' 1 'JUSQUA' M 'FAIRE'

ECRIRE(,,, A.(J,1), A.(J,2), A.(J,3), A.(J,4), A.(J,5),

A.(J,6), A.(J,7), A.(J,8).) FF 'ALLERA' END FF

ZUTF ECRIRE('ARRETEZ CA ME F', 'ATIGUE') FF

ENDF

SAUTLIGNE FF SAUTLIGNE FF SAUTLIGNE FF

'FIN' IMPMAT FF

'PROCEDURE' SURELAXIGMA(B,Y,EPS,OME,N) FF 'TABLEAU' B,Y FF

'REEL' EPS,OME FF 'ENTIER' N FF

'COMMENTAIRE' CETTE PROCEDURE RESOUD LE SYSTEME SIGMA*Y=U PAR

UNE SURELAXATION CONDUITE JUSQU'A CE QUE SIGMA*Y-U AIT UNE

NORME INFINIE INFERIEURE A EPS. SIGMA ET U NE SONT PAS

CONSTRUITS EFFECTIVEMENT ET Y SOLUTION D'UN SYSTEME D'ORDRE

N(N-1)/2 SERA RANGE DANS UN TABLEAU (N,N), LA MATRICE

CORRESPONDANT A Y ETANT ANTISYMETRIQUE FF

'DEBUT' 'TABLEAU' R.(1FN,1FN). FF 'ENTIER' I,J,K,L FF

'REEL' NOR,S,PHINF,CL,TR,MOME,UR FF

PHINFF=0.0 FF MOMEF=1-OME FF

RESIDUS INITIAUXF

'POUR' KF=1 'PAS' 1 'JUSQUA' N 'FAIRE'

'POUR' LF=K+1 'PAS' 1 'JUSQUA' N 'FAIRE'

'DEBUT' SF=0.0 FF

'POUR' IF=L+1 'PAS' 1 'JUSQUA' N 'FAIRE'

SF=S-Y.(I,L).*B.(K,I). FF

```

'POUR' JF=1 'PAS' 1 'JUSQUA' K-1 'FAIRE'
  SF=S+Y.(K,J).*B.(L,J). FF
'POUR' IF=K+1 'PAS' 1 'JUSQUA' L-1,L+1 'PAS' 1
'JUSQUA' N 'FAIRE'
  SF=S-Y.(I,K).*B.(L,I). FF
'POUR' JF=1 'PAS' 1 'JUSQUA' K-1,K+1 'PAS' 1
'JUSQUA' L-1 'FAIRE'
  SF=S+Y.(L,J).*B.(K,J). FF
  SF=S+Y.(L,K).(B.(K,K).-B.(L,L).) FF
  R.(L,K).F=S+B.(L,K). FF
  'SI' ABS(R.(L,K).) 'SUPER' PHINF 'ALORS'
  PHINFF=ABS(R.(L,K).) FF
'FIN' FF
ITKF KF=1 FF
ITLF LF=K+1 FF
TESTF 'SI' L=N+1 'ALORS'
'DEBUT' 'SI' K=N 'ALORS' 'ALLERA' ITK FF
  KF=K+1 FF 'ALLERA' ITL FF
'FIN'
'SINON'
'DEBUT' RELAXATION SUR YLKF
CLF=R.(L,K)./(B.(K,K).-B.(L,L).) FF TRF=Y.(L,K). FF
Y.(L,K).F=TR-CL FF Y.(L,K).F=OME*Y.(L,K).+MOME*TR FF
R.(L,K).F=MOME*R.(L,K). FF PHINF F= 0.0 FF
CALCUL DES NOUVEAUX RESIDUSF
'POUR' JF=1 'PAS' 1 'JUSQUA' K-1,K+1 'PAS' 1
'JUSQUA' L-1 'FAIRE'
'DEBUT' TRF=R.(L,J). FF URF=TR-B.(J,K).*CL FF
  NORF=R.(L,J).F=OME*UR+MOME*TR FF
  'SI' ABS(NOR) 'SUPER' PHINF 'ALORS'
  PHINFF=ABS(NOR) FF
'FIN' FF
'POUR' JF=1 'PAS' 1 'JUSQUA' K-1 'FAIRE'
'DEBUT' TRF=R.(K,J). FF URF=TR+B.(J,L).*CL FF
  NORF=R.(K,J).F=OME*UR+MOME*TR FF
  'SI' ABS(NOR) 'SUPER' PHINF 'ALORS' PHINFF=ABS(NOR) FF
'FIN' FF
'POUR' IF=K+1 'PAS' 1 'JUSQUA' L-1,L+1 'PAS' 1
'JUSQUA' N 'FAIRE'
'DEBUT' TRF=R.(I,K). FF URF=TR+B.(I,L).*CL FF
  NORF=R.(I,K).F=OME*UR+MOME*TR FF
  'SI' ABS(NOR) 'SUPER' PHINF 'ALORS' PHINFF=ABS(NOR) FF
'FIN' FF
'POUR' IF=L+1 'PAS' 1 'JUSQUA' N 'FAIRE'
'DEBUT' TRF=R.(I,L). FF URF=TR-B.(I,K).*CL FF
  NORF=R.(I,L).F=OME*UR+MOME*TR FF
  'SI' ABS(NOR) 'SUPER' PHINF 'ALORS' PHINFF=ABS(NOR) FF
'FIN' FF
'SI' PHINF 'INFER' EPS 'ALORS' 'ALLERA' SORTIE 'SINON'
'DEBUT' LF=L+1 FF 'ALLERA' TEST FF 'FIN' FF
'FIN' RELAXATION FF
SORTIEF 'POUR' IF=1 'PAS' 1 'JUSQUA' N 'FAIRE'
'POUR' JF=I+1 'PAS' 1 'JUSQUA' N 'FAIRE'
Y.(I,J).F=-Y.(J,I). FF
'FIN' PROC RELAX FF
'PROCEDURE' QR(MAT,U,N) FF 'TABLEAU' MAT,U FF 'ENTIER' N FF
'COMMENTAIRE' CETTE PROCEDURE REALISE LA DECOMPOSITION DE LA
MATRICE MAT SOUS LA FORME U*R,U UNITAIRE,R TRIANGULAIRE
SUPERIEURE. R N'EST PAS CONSERVEE FF

```



```

'DEBUT' 'ENTIER' I,J,K,L FF 'REEL' S FF 'TABLEAU' T.(1FN). FF
'POUR' KF=1 'PAS' 1 'JUSQUA' N-1 'FAIRE'
  'DEBUT' SF=0.0 FF
  'POUR' LF=1 'PAS' 1 'JUSQUA' N 'FAIRE'
    SF=S+MAT.(L,K).*MAT.(L,K). FF
  T.(K).F=S FF
  'POUR' LF=1 'PAS' 1 'JUSQUA' K 'FAIRE'
    'DEBUT' SF=0.0 FF 'POUR' JF=1 'PAS' 1 'JUSQUA' N 'FAIRE'
      SF=S+MAT.(J,K+1).*MAT.(J,L). FF
    SF=-S/T.(L). FF
  'POUR' JF=1 'PAS' 1 'JUSQUA' N 'FAIRE'
    MAT.(J,K+1).F=MAT.(J,K+1).+S*MAT.(J,L). FF
  'FIN' FF
'FIN' FF
'POUR' JF=1 'PAS' 1 'JUSQUA' N 'FAIRE'
  'DEBUT' SF=0.0 FF
  'POUR' IF=1 'PAS' 1 'JUSQUA' N 'FAIRE'
    SF=S+MAT.(I,J).*MAT.(I,J). FF
  'POUR' IF=1 'PAS' 1 'JUSQUA' N 'FAIRE'
    U.(I,J).F=MAT.(I,J)./RAC2(S) FF
  'FIN' FF
'FIN' QR FF
'PROCEDURE' ITERQ(CQ,CY,N) FF 'ENTIER' N FF
  'TABLEAU' CQ,CY FF
  'COMMENTAIRE' CETTE PROCEDURE CALCULE LA MATRICE
  Q=Q*(I+YY) FF
  'DEBUT' 'ENTIER' I,J,KFF 'TABLEAU' LIN.(1FN,1FN).FF 'REEL' STFF
  'POUR' IF=1 'PAS' 1 'JUSQUA' N 'FAIRE'
  'POUR' JF=1 'PAS' 1 'A' N 'FAIRE'
    'DEBUT' STF=0 FF
    'POUR' KF=1 'PAS' 1 'JUSQUA' N 'FAIRE'
      'SI' K = J 'ALORS' ST F= ST + CQ.(I,K). 'SINON'
        ST F= ST + CQ.(I,K).*CY.(J,K). FF
      LIN.(I,J). F= ST FF
    'FIN' FF
  'POUR' IF=1 'PAS' 1 'A' N 'FAIRE'
    'POUR' JF=1 'PAS' 1 'JUSQUA' N 'FAIRE'
      CQ.(I,J). F= LIN.(I,J). FF
  'FIN' ITERQ FF
'PROCEDURE' QTAQ(GA,GQ,GB,N) FF 'TABLEAU' GA,GQ,GB FF
  'ENTIER' N FF
  'COMMENTAIRE' CETTE PROCEDURE CALCULE LE PRODUIT
  QT*A*Q=B. FF
  'DEBUT' 'ENTIER' I,J,K,P FF 'REEL' S,T FF
  'POUR' IF=1 'PAS' 1 'JUSQUA' N 'FAIRE'
  'POUR' JF=1 'PAS' 1 'JUSQUA' N 'FAIRE'
  'DEBUT' SF=0.0 FF
  'POUR' KF=1 'PAS' 1 'JUSQUA' N 'FAIRE'
  'DEBUT' TF=0.0 FF
  'POUR' PF=1 'PAS' 1 'JUSQUA' N 'FAIRE'
    T F= T+GA.(K,P).*GQ.(P,J). FF
    S F= S + GQ.(K,I). * T FF
  'FIN' FF
  GB.(I,J). F= S FF
  'FIN' FF
  'FIN' QTAQ FF
'PROCEDURE' NEWTONMAT(A,Q,B,EPS,ETA,N) FF 'TABLEAU' A,Q,B FF
  'REEL' EPS,ETA FF 'ENTIER' N FF
  'COMMENTAIRE' CETTE PROCEDURE RECHERCHE LA FORME DIAGONALE DE
  A,MATRICE SYMETRIQUE. ON CONSTRUIT UNE SUITE DE MATRICES QP

```

```

TELLES QUE T(QP)*A*QP TENDE VERS LA FORME DIAGONALE DE A FF
'DEBUT' 'ENTIER' I,J,K,L,CT FF
'REEL' 'TABLEAU' Y.(1FN,1FN). FF 'REEL' PREC,OME FF
'REEL' 'PROCEDURE' NORMND FF
'DEBUT' 'ENTIER' I,J FF 'REEL' D FF
DF=0.0 FF
'POUR' IF=1 'PAS' 1 'JUSQUA' N 'FAIRE'
'POUR' JF=1 'PAS' 1 'JUSQUA' I-1,I+1 'PAS' 1 'JUSQUA' N 'FAIRE'
DF=D+B.(I,J).*B.(I,J). FF
NORMNDF=RAC2(D) FF
'FIN' NORMND FF
INITIALISATIONSF
'POUR' I F= 1 'PAS' 1 'A' N 'FAIRE'
'POUR' J F= 1 'PAS' 1 'A' N 'FAIRE'
B.(I,J). F= A.(I,J). FF
CTF=0 FF
RECURF
'POUR' I F= 1 'PAS' 1 'A' N 'FAIRE'
'POUR' J F= I + 1 'PAS' 1 'A' N 'FAIRE'
'DEBUT' Y.(I,J). F= 0.1FF
Y.(J,I). F= -0.1FF
'FIN' FF
OME F= 1.0 FF
SURELAXIGMA(B,Y,ETA,OME,N) FF
ITERQ(Q,Y,N) FF
QR(Q,Q,N) FF
QTAQ(A,Q,B,N) FF
Ecrire('MATRICE B') FF IMPMAT(B,N,N) FF
PRECF=NORMND FF
'SI' PREC 'INFER' EPS 'ALORS' 'ALLERA' SORT 'SINON'
'DEBUT' CTF=CT+1 FF
'SI' CT 'SUPER'20 'ALORS' 'ALLERA' IMPRIME FF
'ALLERA' RECUR FF
'FIN' FF
IMPRIMEF Ecrire('PAS DE CONVERGE', 'INCE APRES',
CT-1, 'ITERATIONS') FF 'ALLERA' FIN FF
SORTF Ecrire('NORME DES ELEME', 'INTS NON DIAGONA',
'UX DE B=', NORMND) FF
FINF IMPMAT(B,N,N) FF IMPMAT(Q,N,N) FF
'FIN' NEWTONMAT FF
'ENTIER' N,I,J FF 'REEL' EPS,ETA FF LIRE(N,EPS,ETA) FF
'DEBUT' 'TABLEAU' AQ,AA,AB.(1FN,1FN). FF
'POUR' IF=1 'PAS' 1 'JUSQUA' N 'FAIRE'
'POUR' JF=1 'PAS' 1 'JUSQUA' N 'FAIRE'
AA.(I,J).F=RDONNEE FF
'POUR' IF=1 'PAS' 1 'JUSQUA' N 'FAIRE'
'POUR' JF=1 'PAS' 1 'JUSQUA' N 'FAIRE'
AQ.(I,J).F= 'SI' I=J 'ALORS' 1 'SINON' 0 FF
Ecrire('MATRICE DONNEE') FF IMPMAT(AA,N,N) FF
NEWTONMAT(AA,AQ,AB,EPS,ETA,N) FF
'FIN' FF
'FIN' DU PROGRAMME FF

```