

50376
1983
151

N° d'ordre : 1048

50376
1983
151

UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

THÈSE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

pour obtenir le titre de

DOCTEUR DE 3^{ème} CYCLE

«Mathématiques Appliquées»

par

Daniel MOUKOKO



**CONCEPTION DES LOGICIELS SUR L'ALGORITHME
QD GENERAL.APPLICATIONS**

Thèse soutenue le 17 Juin 1983 devant la Commission d'Examen

Membres du Jury :

Président	P. POUZET
Rapporteur	A. DRAUX
Examineur	C. BREZINSKI

PROFESSEURS 1ère CLASSE

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

M. MAILLET Pierre	S.E.S.
M. MONTREUIL Jean	Biologie
M. PAQUET Jacques	Sciences de la Terre
M. PARREAU Michel	Mathématiques
M. PROUVOST Jean	Sciences de la Terre
M. SALMER Georges	I.E.E.A.
Mme SCHWARTZ Marie-Hélène	Mathématiques
M. SEGUIER Guy	I.E.E.A.
M. STANKIEWICZ François	Sciences Economiques
M. TILLIEU Jacques	Physique
M. TRIDOT Gabriel	Chimie
M. VIDAL Pierre	I.E.E.A.
M. VIVIER Emile	Biologie
M. WERTHEIMER Raymond	Physique
M. ZEYTOUNIAN Radyadour	Mathématiques

PROFESSEURS 2ème CLASSE

M. AL FAKIR Sabah	Mathématiques
M. ANTOINE Philippe	Mathématiques
M. BART André	Biologie
Mme BATTIAU Yvonne	Géographie
M. BEGUIN Paul	Mathématiques
M. BELLET Jean	Physique
M. BKOUCHE Rudolphe	Mathématiques
M. BOBE Bernard	S.E.S.
M. BODART Marcel	Biologie
M. BOILLY Bénoni	Biologie
M. BONNELLE Jean-Pierre	Chimie
M. BOSQ Denis	Mathématiques
M. BREZINSKI Claude	I.E.E.A.
M. BRUYELLE Pierre (Chargé d'enseignement)	Géographie
M. CAPURON Alfred	Biologie
M. CARREZ Christian	I.E.E.A.
M. CHAMLEY Hervé	E.U.D.I.L.

M. CHAPOTON Alain	C.U.E.E.P.
M. COQUERY Jean-Marie	Biologie
Mme CORSIN Paule	Sciences de la Terre
M. CORTOIS Jean	Physique
M. COUTURIER Daniel	Chimie
Mle DACHARRY Monique	Géographie
M. DEBRABANT Pierre	E.U.D.I.L.
M. DEGAUQUE Pierre	I.E.E.A.
M. DELORME Pierre	Biologie
M. DEMUNTER Paul	C.U.E.E.P.
M. DE PARIS Jean-Claude	Mathématiques
M. DEVRAINNE Pierre	Chimie
M. DHAINAUT André	Biologie
M. DORMARD Serge	S.E.S.
M. DOUKHAN Jean-Claude	E.U.D.I.L.
M. DUBOIS Henri	Physique
M. DUBRULLE Alain	Physique
M. DUEE Gérard	Sciences de la Terre
M. DYMENT Arthur	Mathématiques
M. FLAMME Jean-Marie	E.U.D.I.L.
M. FONTAINE Hubert	Physique
M. GERVAIS Michel	S.E.S.
M. GOBLOT Rémi	Mathématiques
M. GOSSELIN Gabriel	S.E.S.
M. GOUDMAND Pierre	Chimie
M. GREVET Patrice	S.E.S.
M. GUILBAULT Pierre	Biologie
M. HANGAN Théodore	Mathématiques
M. HERMAN Maurice	Physique
M. JACOB Gérard	I.E.E.A.
M. JACOB Pierre	Mathématiques
M. JOURNEL Gérard	E.U.D.I.L.
M. KREMBEL Jean	Biologie
M. LAURENT François	I.E.E.A.
Mle LEGRAND Denise	Mathématiques
Mle LEGRAND Solange	Mathématiques (Calais)
Mme LEHMANN Josiane	Mathématiques

M. LEMAIRE Jean	Physique
M. LENTACKER Firmin	G.A.S.
M. LEVASSEUR Michel	I.P.A.
M. LHENAFF René	G.A.S.
M. LOCQUENEUX Robert	Physique
M. LOSFELD Joseph	I.E.E.A.
M. LOUAGE Francis	E.U.D.I.L.
M. MACKE Bruno	Physique
M. MAIZIERES Christian	I.E.E.A.
Mlle MARQUET Simone	Mathématiques
M. MESSELYN Jean	Physique
M. MIGEON Michel	E.U.D.I.L.
M. MIGNOT Fulbert	Mathématiques
M. MONTEL Marc	Physique
Mme NGUYEN VAN CHI Régine	G.A.S.
M. PARSY Fernand	Mathématiques
Mlle PAUPARDIN Colette	Biologie
M. PERROT Pierre	Chimie
M. PERTUZON Emile	Biologie
M. PONSOLLE Louis	Chimie
M. PORCHET Maurice	Biologie
M. POVY Lucien	E.U.D.I.L.
M. RACZY Ladislas	I.E.E.A.
M. RICHARD Alain	Biologie
M. RIETSCH François	E.U.D.I.L.
M. ROGALSKI Marc	M.P.A.
M. ROUSSEAU Jean-Paul	Biologie
M. ROY Jean-Claude	Biologie
M. SALAMA Pierre	S.E.S.
Mme SCHWARZBACH Yvette (CCP)	M.P.A.
M. SCHAMPS Joël	Physique
M. SIMON Michel	S.E.S.
M. SLIWA Henri	Chimie
M. SOMME Jean	G.A.S.
Mlle SPIK Geneviève	Biologie
M. STERBOUL François	E.U.D.I.L.
M. TAILLIEZ Roger	Institut Agricole

M. TOULOTTE Jean-Marc	I.E.E.A.
M. VANDORPE Bernard	E.U.D.I.L.
M. WALLART Francis	Chimie
M. WATERLOT Michel	Sciences de la Terre
Mme ZINN JUSTIN Nicole	M.P.A.

CHARGES DE COURS

M. TOP Gérard	S.E.S.
M. ADAM Michel	S.E.S.

CHARGES DE CONFERENCES

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

THESE DE 3^{ème} CYCLE

CONCEPTION DES LOGICIELS SUR L'ALGORITHME QU GENERAL,
APPLICATIONS

Daniel MOUKOKO

UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE I
U.E.R. d'I.E.E.A. - Service Informatique
59655 VILLENEUVE D'ASCO CEDEX

☆☆☆☆☆
 ☆☆☆
 ☆☆
 ☆

Thèse soutenue le 17 Juin 1983 devant la Commission d'Examen.

Membres du Jury : Président : P. POUZET
Rapporteur : A. DRAUX
Examineur : C. BREZINSKI

REMERCIEMENTS

Monsieur le Professeur P. POUZET a bien voulu me faire l'honneur de présider ce Jury. Je le prie de trouver ici l'expression de ma gratitude.

Monsieur A. DRAUX, Docteur d'Etat, est à l'origine de ce travail qu'il a dirigé et suivi. Ses conseils, ses encouragements et son soutien m'ont été très précieux. Je lui exprime ma profonde reconnaissance.

Monsieur le Professeur C. BREZINSKI a suivi l'évolution de ce travail, ses remarques et l'intérêt qu'il a porté à ce travail m'ont été fort utiles. Je suis très sensible à l'honneur qu'il me fait en participant à ce Jury.

Enfin, tous mes remerciements à Mademoiselle FIEVET Bénédicte qui, avec gentillesse et rapidité, a tapé cette thèse et à Mr et Mme DEBOCK qui en ont assuré le tirage.

TABLE DES MATIERES

Pages

1^{ère} Partie : RAPPELS2^{ème} Partie : ALGORITHME qd GENERAL

- I. INTRODUCTION
- II. ALGORITHME DE CALCUL
 - A) Méthode normale
 - B) Méthode progressive
- III. MISE EN OEUVRE
 - Les sous-programmes

3^{ème} Partie : POLYNOMES ORTHOGONAUX ET APPROXIMANTS DE PADE

- I. SYSTEMES ADJACENTS DES POLYNOMES ORTHOGONAUX ET LEURS ASSOCIES
 - Algorithme de calcul
- II. APPROXIMANTS DE PADE
- III. MISE EN OEUVRE
 - Les sous-programmes

4^{ème} Partie : FONCTIONNELLES LINEAIRES LACUNAIRES D'ORDRE $s+1$

- I. DEFINITION ET RAPPELS
- II. MISE EN OEUVRE
 - Les sous-programmes

5^{ème} Partie : APPROXIMANTS DE PADE EN DEUX POINTS

- I. DEFINITIONS ET RAPPELS
- II. MISE EN OEUVRE
 - Les sous-programmes

6^{ème} Partie : APPROXIMANTS DES SERIES DE FONCTIONS

- I. DEFINITIONS ET RAPPELS
- II. CALCUL DES RACINES DES POLYNOMES
- III. MISE EN OEUVRE
 - Les sous-programmes

7^{ème} Partie : ALGORITHME $\tilde{q}d$

- I. DEFINITION ET RELATION AVEC L'ALGORITHME qd
- II. MISE EN OEUVRE
 - Le sous-programme QDTILD

8^{ème} Partie : SOUS-PROGRAMMES EN ARITHMETIQUE RATIONNELLE

- VIII.1. INTRODUCTION
- VIII.2. SOUS-PROGRAMMES DE CALCUL DE LA TABLE qd PAR
LES METHODES NORMALE ET PROGRESSIVE
- VIII.3. SOUS-PROGRAMMES DE CALCUL DES POLYNOMES ORTHO-
GONAUX REGULIERS ET LEURS ASSOCIES, ET DES
APPROXIMANTS DE PADE
- VIII.4. SOUS-PROGRAMMES DE CALCUL DES APPROXIMANTS DE
PADE EN DEUX POINTS
- VIII.5. SOUS-PROGRAMME DE CALCUL DES APPROXIMANTS DES
SERIES DE FONCTIONS

9^{ème} Partie : COMPARAISONS ET CONCLUSION

INTRODUCTION

Le calcul des éléments de la table qd , dans le cas général, a été rendu possible grâce à l'algorithme qd généralisé par DRAUX [1]. Il l'appelle algorithme qd général pour le distinguer du classique.

Notre travail a consisté à la mise en oeuvre des logiciels Fortran qui calculent les éléments de la table qd en utilisant l'algorithme qd général. Le calcul de la table qd s'effectue au choix, par la méthode directe (ou normale) ou par la méthode progressive.

Nous proposons des sous-programmes complémentaires qui calculent les polynômes orthogonaux réguliers et leurs associés, les approximants de Padé en un point et en deux points, et les approximants des séries de fonctions.

Les résultats sont obtenus pour des situations absolument quelconques en arithmétique réelle double précision ou en arithmétique rationnelle étendue (arithmétique exacte de longueur de mot arbitraire).

Nous abordons aussi l'algorithme $\tilde{qd}$, que l'on définit à partir des relations de récurrence à trois termes le long des antidiagonales, et le calcul des racines des polynômes. Nous nous sommes aussi intéressés au calcul des polynômes orthogonaux lacunaires.

Ce travail se divise en neuf parties :

La première partie est consacrée aux définitions et à certains résultats classiques qui sont généralisés dans [1], qui servent dans la suite.

Dans la deuxième nous proposons un algorithme de calcul des éléments de la table qd et une mise en oeuvre de ce dernier.

Dans les troisième et quatrième nous donnons un algorithme qui calcule les polynômes orthogonaux réguliers (lacunaires pour la quatrième) et leurs associés, et proposons des sous-programmes qui calculent les polynômes orthogonaux réguliers, leurs associés et les approximants de Padé.

Dans la cinquième nous donnons un ensemble de sous-programmes calculant les approximants de Padé en deux points.

La sixième est consacrée au calcul des approximants des séries de fonctions et au calcul des racines des polynômes.

Dans la septième nous montrons comment on peut déduire l'algorithme $\tilde{q}d$ à partir de l'algorithme qd .

Dans l'avant dernière partie nous reprenons les sous-programmes qui calculent :

- les éléments de la table qd ,
- les polynômes orthogonaux réguliers, leurs associés et les approximants de Padé,
- les approximants de Padé en deux points,
- les approximants des séries de fonctions

en arithmétique rationnelle étendue.

Enfin la dernière partie est consacrée à la comparaison des résultats des arithmétiques réelle double précision et rationnelle étendue sur un exemple, et à la conclusion.

Notre travail a consisté à résoudre quelques-uns des points soulevés par DRAUX dans son livre [1].

N.B. Nos programmes ont été mis au point avec un ordinateur qui permet de travailler avec les indices négatifs (IRIS 80), et donc peuvent susciter des modifications pour passer dans certains ordinateurs.

PREMIERE PARTIE

I. RAPPELS

Cette partie rappelle les définitions et certains résultats classiques, qui sont généralisés dans [1], qui vont servir dans toute la suite.

Définition 1.1. : Soit une suite de nombres réels $(c_i)_{i \in \mathbb{N}}$, le déterminant

$$\begin{vmatrix} c_n & c_{n+1} & & c_{n+k-1} \\ & c_{n+1} & & \\ & & & \\ & & & \\ c_{n+k-1} & & & c_{n+2k-2} \end{vmatrix}$$

est appelé déterminant de Hankel et est noté $H_k^{(n)}$.

Par convention on posera $H_0^{(n)} = H_{-1}^{(n)} = 1 \quad \forall n \in \mathbb{N}$.

Définition 1.2. : On appelle table H le tableau triangulaire contenant les $H_k^{(n)}$

$$\begin{array}{cccc} H_{-1}^{(0)} & & & \\ H_{-1}^{(1)} & H_0^{(0)} & & \\ H_{-1}^{(2)} & H_0^{(1)} & H_1^{(0)} & \\ H_{-1}^{(3)} & H_0^{(2)} & H_1^{(1)} & H_2^{(0)} \\ \vdots & \vdots & \vdots & \ddots \\ \vdots & \vdots & \vdots & \ddots \\ \vdots & \vdots & \vdots & \ddots \end{array}$$

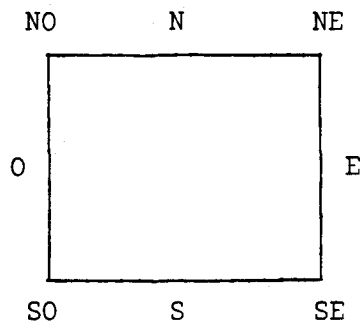
où l'indice supérieur correspond à l'indice de la diagonale et l'autre à l'indice de la colonne.

On dit que la table H est normale si $H_k^{(i)} \neq 0 \quad \forall i, k \in \mathbb{N}$, sinon elle est dite non normale.

Définition 1.3. : On appelle bloc de la table H un bloc où tous les déterminants sont nuls.

Remarques 1.4. : La table H ne contient que des blocs carrés de zéro.

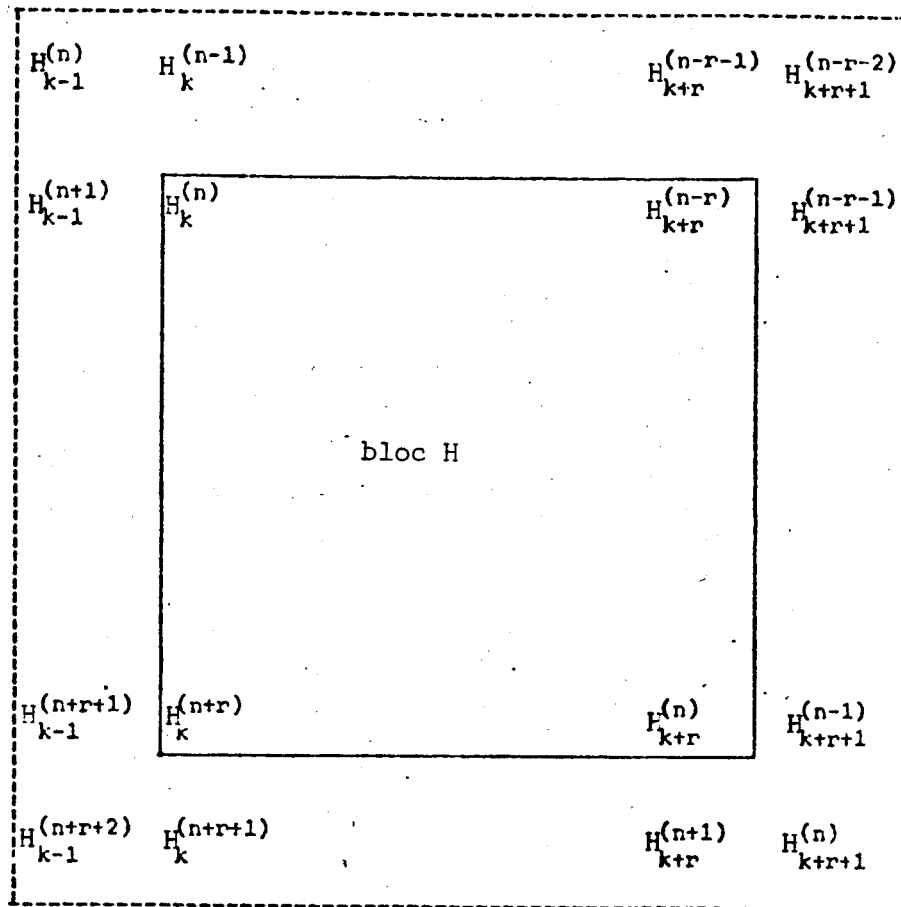
On définit les côtés et les coins d'un bloc H par un repère géographique (Nord, Sud, Ouest et Est) pour les côtés, (Nord-Ouest, Nord-Est, Sud-Ouest et Sud-Est) pour les coins.



Les blocs seront aussi définis par des triplets (n, k, r) où :

n est l'indice de la diagonale principale,
 k l'indice de la première colonne du bloc
 et r tel que $r+1$ soit la largeur du bloc.

Le bloc (n, k, r) de la table H est de la forme :



Définition 1.5. : Soient P l'espace vectoriel des polynômes réels et (c_i) $i \in \mathbb{N}$ une suite de nombres réels. On peut définir une fonctionnelle linéaire c sur P telle que

$$c(x^i) = c_i \quad \forall i \in \mathbb{N}.$$

Notons P_j un polynôme de degré j exactement.

Définition 1.6. : Le polynôme P_k est dit orthogonal par rapport à une fonctionnelle linéaire c si $c(P_k P_i) = 0 \quad \forall i \in \mathbb{N}, i < k$.

Définition 1.7. : Un polynôme est dit unitaire si le coefficient de son terme de plus grand degré est égal à 1.

Remarque 1.8. : Dans toute la suite P_k sera un polynôme unitaire de degré k exactement.

Théorème 1.9. : Une condition suffisante pour que le polynôme P_k orthogonal par rapport à une fonctionnelle linéaire c existe est que $H_k^{(0)} \neq 0$. Alors il est unique.

Faisons l'hypothèse $H_k^{(0)} \neq 0 \quad \forall k \in \mathbb{N}$, on a :

Propriété 1.10. : $P_0(x) = 1$

$$P_k(x) = \frac{\begin{vmatrix} c_0 & & & & c_k \\ & c_1 & & & \vdots \\ & \vdots & & & \vdots \\ & \vdots & & & \vdots \\ & c_{k-1} & & & c_{2k-1} \\ & 1 & & & x^k \end{vmatrix}}{H_k^{(0)}}$$

Théorème 1.11. : Les polynômes orthogonaux (P_k) vérifient la relation de récurrence à trois termes suivante :

$$P_{k+1}(x) = (x + B_{k+1}) P_k(x) + C_{k+1} P_{k-1}(x)$$

$$\text{avec } C_{k+1} = -c(x P_{k-1}(x) P_k(x)) / c(P_{k-1}^2(x))$$

$$\text{et } B_{k+1} = -c(x P_k^2(x)) / c(P_k^2(x))$$

On démarre cette relation avec $P_{-1}(x) = 0$ et $P_0(x) = 1$.

Définition 1.12. : On appellera systèmes adjacents des polynômes orthogonaux $\{P_k^{(n)}\}$ $k \in \mathbb{N}$ les polynômes orthogonaux par rapport aux fonctionnelles linéaires $c^{(n)}$ définies par

$$c^{(n)}(x^j) = c_{n+j} \quad \forall j \in \mathbb{N}, \forall n \in \mathbb{N}$$

Une condition suffisante pour que $P_k^{(n)}$ existe est que $H_k^{(n)} \neq 0$. Alors il est unique.

Définition 1.13. : On appelle table P la table des polynômes $\{P_k^{(n)}\}$ écrite sous forme d'un tableau triangulaire :

$$P_{-1}^{(0)}$$

$P_{-1}^{(1)}$

$$P_0^{(0)}$$

$$P_{-1}^{(2)}$$

$$P_0^{(1)}$$

$$P_1(0)$$

•
•
•
•
•
•

•

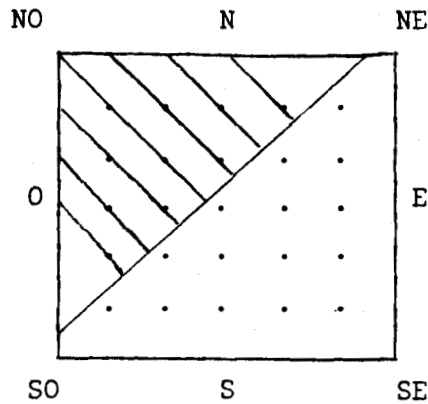
•

•

•

- .

- 1



l'autre partie est occupée par les polynômes quasi-orthogonaux.

Définition 1.19. : Soit $\{P_k^{(n)}\}$ l'ensemble des systèmes adjacents des polynômes orthogonaux par rapport aux $c^{(n)}$. On appellera polynômes associés aux polynômes orthogonaux les polynômes définis par

$$Q_k^{(n)}(t) = c^{(n)} \left(\frac{P_k^{(n)}(x) - P_k^{(n)}(t)}{x - t} \right),$$

avec $c^{(n)}$ agissant sur la variable x .

Remarque 1.20. : Les polynômes associés aux polynômes orthogonaux vérifient la même relation de récurrence à trois termes que les polynômes orthogonaux, avec au démarrage $Q_{-1}^{(n)}(t) = 1$ et $Q_0^{(n)}(t) = 0 \quad \forall n \in \mathbb{N}$.

Définition 1.21. : Soient la série formelle $f(t) = \sum_{j=0}^{\infty} c_j t^j$ et, $A(t)$ et $B(t) \neq 0$ deux polynômes de degrés a et b . La fraction rationnelle $\frac{A(t)}{B(t)}$ est appelée approximant de Padé et est noté $[a/b]_f(t)$

$$\text{si } f(t) - \frac{A(t)}{B(t)} = O(t^{a+b+1}).$$

Théorème 1.22. : Soit $f(t) = \sum_{j=0}^{\infty} c_j t^j$ alors si $P_k^{(n)}$ existe on a :

$$[k-1/k]_{f_n} = \frac{\gamma_k^{(n)}(t)}{P_k^{(n)}(t)}$$

$$\text{avec } \gamma_k^{(n)}(t) = t^{k-1} Q_k^{(n)}(t^{-1}), \quad \gamma_k^{(n)}(t) = t^k P_k^{(n)}(t^{-1})$$

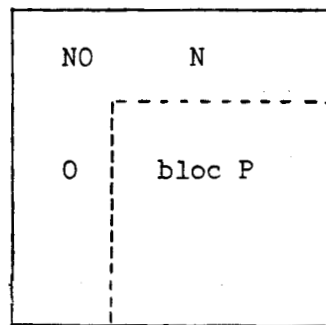
$$\text{et } f_n(t) = \sum_{j=0}^{\infty} c_{n+j} t^j.$$

Corollaire 1.23. : $[n+k-1/k]_f(t) = \sum_{j=0}^{n-1} c_j t^j + t^n [k-1/k]_{f_n}(t)$ avec $n+k-1 \geq 0$

Définition 1.24. : On appellera table de Padé la table des approximants de Padé $[n/k]_f(t)$ écrite sous la forme d'un tableau rectangulaire

[0/0]	[0/1]	[0/2]	[0/3]
[1/0]	[1/1]	[1/2]	[1/3]
[2/0]	[2/1]	[2/2]	[2/3]
⋮	⋮	⋮	⋮

Définition 1.25. : On appellera bloc de la table de Padé la réunion de la partie qui correspond au bloc P et des parties correspondant aux côtés Nord, Nord-Ouest et Ouest de ce dernier.



Supposons $H_k^{(n)} \neq 0 \quad \forall k, n \in \mathbb{N}$, on sait que les polynômes orthogonaux $\{P_k^{(n)}\}$ existent et vérifient la relation de récurrence à trois termes

$$P_{k+1}^{(n)}(x) = (x + B_{k+1}^{(n)}) P_k^{(n)}(x) + C_{k+1}^{(n)} P_{k-1}^{(n)}(x)$$

avec $\begin{cases} P_{-1}^{(n)}(x) = 0 \\ P_0^{(n)}(x) = 1 \end{cases} \quad \forall n \in \mathbb{N}.$

Si on pose $B_{k+1}^{(n)} = -q_{k+1}^{(n)} - e_k^{(n)}$ et $C_{k+1}^{(n)} = -e_k^{(n)} q_k^{(n)}$ on a les résultats suivants :

Propriété 1.26. : Les polynômes orthogonaux vérifient les relations

$$P_{k+1}^{(n)}(x) = x P_k^{(n+1)}(x) - q_{k+1}^{(n)} P_k^{(n)}(x)$$

et $P_k^{(n+1)}(x) = P_k^{(n)}(x) - e_k^{(n)} P_{k-1}^{(n+1)}(x)$, avec $e_0^{(n)} = 0$ et la convention $q_0^{(n)} e_0^{(n)} = c_n$, que l'on appellera relations de récurrence sur deux diagonales adjacentes.

On démontre que les polynômes $Q_k^{(n)}(t)$ associés aux polynômes orthogonaux vérifient les relations suivantes :

$$Q_{k+1}^{(n)}(t) = Q_k^{(n+1)}(t) - q_{k+1}^{(n)} Q_k^{(n)}(t) + c_n P_k^{(n+1)}(t)$$

$$\text{et } Q_k^{(n+1)}(t) = t Q_k^{(n)}(t) - c_n P_k^{(n)}(t) - e_k^{(n)} Q_{k-1}^{(n+1)}(t), \text{ avec } e_0^{(n)} = 0$$

et la convention $q_0^{(n)} e_0^{(n)} = c_n$.

Propriété 1.27. : Les $\{q_k^{(n)}\}$ et les $\{e_k^{(n)}\}$ satisfont les relations

$$q_{k+1}^{(n)} + e_{k+1}^{(n)} = q_{k+1}^{(n+1)} + e_k^{(n+1)} \quad (1)$$

$$\text{et } q_{k+1}^{(n)} \times e_k^{(n)} = q_k^{(n+1)} \times e_k^{(n+1)} \quad (2)$$

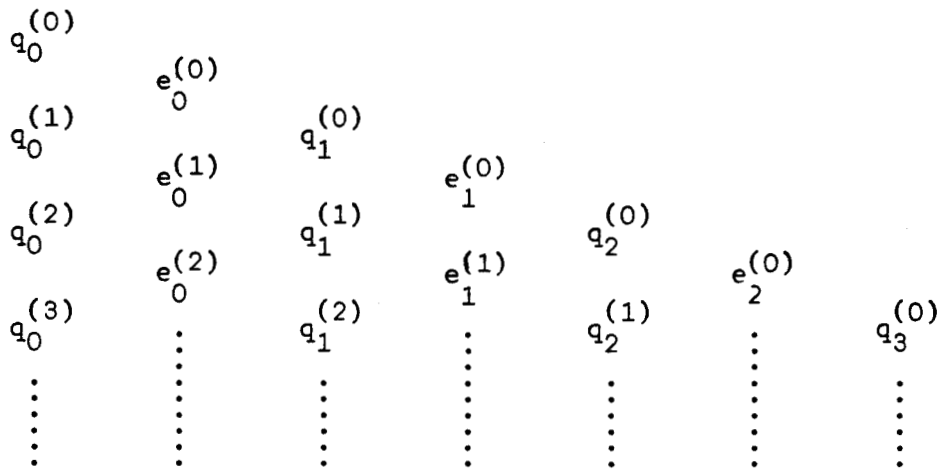
avec au départ $e_0^{(n)} = q_k^{(n)} = 0$ et $q_1^{(n)} = c_{n+1}/c_n$.

Ces relations sont appelées relations rhomboidales.

Définition 1.28. : L'algorithme de calcul des $\{q_k^{(n)}\}$ et $\{e_k^{(n)}\}$ est appelé algorithme qd.

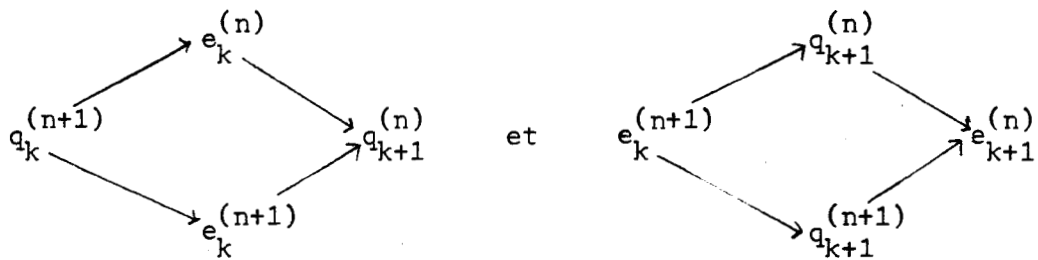
RUTISHAUSER [5] et HENRICI [6] se sont particulièrement intéressés à cet algorithme que DRAUX généralise dans son livre [1].

Définition 1.29. : On appelle table qd un tableau triangulaire dans lequel les $\{q_k^{(n)}\}$ et $\{e_k^{(n)}\}$ sont rangés de la manière suivante :



L'indice supérieur étant l'indice de la diagonale et l'autre celui de la colonne.

Les relations (1) et (2) relient les quantités situées aux quatre sommets de deux losanges :



$q_{k+1}^{(n)}$ et $e_{k+1}^{(n)}$ se calculent facilement si l'on connaît les trois autres éléments du losange.

A un bloc P correspond un bloc de la table q_d , ces deux blocs auront les mêmes coordonnées.

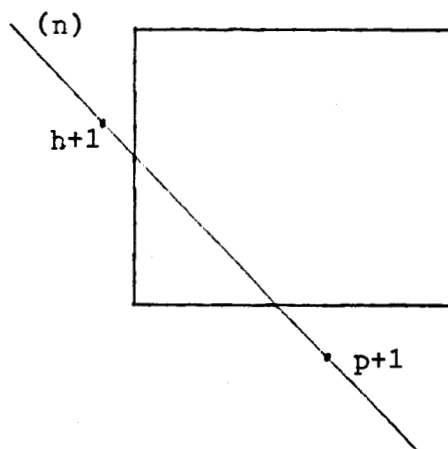
DEUXIEME PARTIE

ALGORITHME qd GENERAL
I. INTRODUCTION

Nous proposons dans cette partie un algorithme de calcul des éléments de la table qd , en utilisant l'algorithme qd général de DRAUX, par les méthodes normale et progressive. Nous proposons aussi un ensemble de sous-programmes Fortran écrits en arithmétique réelle simple ou double précision qui calcule les éléments de la table qd .

NOTATIONS

- On écrira O.R. pour dire orthogonal régulier.
- $P_{pr(i,n)}^{(n)}(x)$ est le polynôme O.R. qui précède le polynôme $P_i^{(n)}(x)$ sur la diagonale n, $pr(i,n)$ est le degré de $P_{pr(i,n)}^{(n)}(x)$.
- γ_i est une fonction qui s'annule pour $i = 0$ et vaut 1 pour les autres valeurs de i .
- Si la diagonale n rencontre un bloc P, on notera h et p les deux entiers tels que $H_{h+1}^{(n)} \neq 0$, $H_{p+1}^{(n)} \neq 0$ et $H_i^{(n)} = 0 \quad \forall i \in \mathbb{N}, h+2 \leq i \leq p$.



On peut munir chaque diagonale n d'un compteur de blocs rencontrés $\ell^{(n)}$ et ajouter $\ell^{(n)}$ comme troisième indice aux éléments de la table q_d . On écrira alors $q_{i,\ell^{(n)}}^{(n)}$ et $e_{i,\ell^{(n)}}^{(n)}$.

On indicera aussi les entiers h et p que l'on vient de définir avec le compteur $\ell^{(n)}$ et on aura

$$H_{\ell^{(n)}-1+1}^{(n)} \neq 0, H_{p_{\ell^{(n)}+1}} \neq 0 \text{ et } H_i^{(n)} = 0 \text{ pour } i \in \mathbb{N}, h_{\ell^{(n)}-1} + 2 \leq i \leq p_{\ell^{(n)}}$$

Nous avons rappelé dans I les relations de récurrence que les polynômes orthogonaux vérifient, dans le cas normal, sur les diagonales et sur deux diagonales adjacentes, celles-ci sont généralisées par les deux propriétés suivantes :

Propriété 2.1. : [1, page 71] : Tout polynôme orthogonal $P_i^{(n)}(x)$ est obtenu à l'aide des deux polynômes orthogonaux réguliers qui le précèdent :

$$P_i^{(n)}(x) = (x \omega_{i-\text{pr}(i,n)-1}^{(n)}(x) + B_i^{(n)}) P_{\text{pr}(i,n)}^{(n)}(x) + C_i^{(n)} P_{\text{pr}(\text{pr}(i,n),n)}^{(n)}(x)$$

avec

- $\omega_{i-\text{pr}(i,n)-1}^{(n)}(x)$ un polynôme unitaire de degré $i-\text{pr}(i,n)-1$
- et
- $P_{\text{pr}(\text{pr}(i,n),n)}^{(n)}(x)$ le polynôme O.R. qui précède le polynôme O.R.

$P_{\text{pr}(i,n)}^{(n)}(x)$ sur la diagonale n .

On démarre cette relation avec $P_{-1}^{(n)}(x) = 0$ et $P_0^{(n)}(x) = 1$.

Propriété 2.2. : [1, page 102] :

- Si le polynôme $P_i^{(n+1)}$ est O.R. alors

$$P_i^{(n+1)}(x) = \omega_{i-\text{pr}(i+1,n)}^{(n)}(x) P_{\text{pr}(i+1,n)}^{(n)}(x) + E_{i+1}^{(n)} P_{\text{pr}(\text{pr}(i+1,n),n+1)}^{(n+1)}(x)$$

où $P_{\text{pr}(\text{pr}(i+1,n),n+1)}^{(n+1)}(x)$ est le polynôme O.R. qui précède $P_{\text{pr}(i+1,n)}^{(n)}$ sur la diagonale $n+1$,

- et si de plus $P_{i+1}^{(n)}(x)$ est O.R. on a :

$$P_i^{(n+1)}(x) = x^{-1}(P_{i+1}^{(n)}(x) + q_{i+1,\ell}^{(n)} P_{pr(i+1,n)}^{(n)}(x))$$

Les $q_{i+1,\ell}^{(n)}$ et $E_{i+1}^{(n)}$ sont définis par :

$$\text{si } p_{\ell}^{(n)} \leq i \leq h_{\ell}^{(n)} \quad B_{i+1}^{(n)} = -q_{i+1,\ell}^{(n)} - e_{pr(i+1,n),\alpha}^{(n)}$$

$$C_{i+1}^{(n)} = -q_{pr(i+1,n),\alpha}^{(n)} e_{pr(i+1,n),\alpha}^{(n)}$$

$$\text{où } \alpha^{(n)} = \ell^{(n)} \text{ si } pr(i+1,n) = i$$

$$= \ell^{(n)} - 1 \text{ sinon}$$

avec au démarrage $e_{0,0}^{(n)} = 0$, $q_{0,0}^{(n)} = 0$ et par convention $-q_{0,0}^{(n)} e_{0,0}^{(n)} = c_n$

$$\text{et } E_{i+1}^{(n)} = \begin{cases} -e_{i,\ell}^{(n)} & \text{si } P_{i-1}^{(n+1)} \text{ est O.R.} \\ C_{i+1}^{(n)} & \text{sinon} \end{cases}$$

Dans certains cas où $q_i^{(n)}$ et $e_i^{(n)}$ sont indéfinis on a :

1 - Si $P_{i+1}^{(n)}$ et $P_i^{(n)}$ sont O.R. et $P_i^{(n+1)}$ est quasi-orthogonal, c'est-à-dire

si $P_i^{(n)}(0) = 0$ et

a) si $P_{h_{\ell}^{(n)}-1+1}^{(n)}(0) \neq 0$ et $p_{\ell}^{(n)} + 1 \leq i \leq h_{\ell}^{(n)}$

ou

b) si $P_{h_{\ell}^{(n)}-1+1}^{(n)}(0) = 0$ et $p_{\ell}^{(n)} + 2 \leq i \leq h_{\ell}^{(n)}$

alors $q_{i,\ell}^{(n)} = 0$, $e_{i,\ell}^{(n)} = \infty$ et $q_{i+1,\ell}^{(n)} = \infty$

2 - Si $P_i^{(n)}(0) = 0$ avec $i < h_{\ell(n)}$, donc si $P_{i+2}^{(n)}$ est O.R.,

$$\text{alors } e_{i+1, \ell(n)}^{(n)} = 0, B_{i+2}^{(n)} = -q_{i+2, \ell(n)}^{(n)}$$

3 - Si $P_{h_{\ell(n)}}^{(n)}(0) = 0$, donc si $P_{h_{\ell(n)}+2}^{(n)}$ est non O.R.

$$\text{alors } P_{h_{\ell(n)}+1}^{(n)}(0) \neq 0, q_{h_{\ell(n)}+1, \ell(n)}^{(n)} = \infty \text{ et } e_{h_{\ell(n)}+1, \ell(n)}^{(n)} = 0$$

4 - Si $P_{h_{\ell(n)}+1}^{(n)}(0) \neq 0$ et $P_{h_{\ell(n)}}^{(n)}(0) \neq 0$, alors $q_{h_{\ell(n)}+1, \ell(n)}^{(n)}$ est fini non nul, $P_{h_{\ell(n)}}^{(n+1)}$ est O.R. et $e_{h_{\ell(n)}+1, \ell(n)}^{(n)} = 0$

5 - Si $P_{h_{\ell(n)}+1}^{(n)}(0) = 0$; c'est-à-dire si $P_{p_{\ell(n)}+1}^{(n)}$ est O.R. et $P_{p_{\ell(n)}+1}^{(n+1)}$ est

quasi-orthogonal alors

$$q_{h_{\ell(n)}+1, \ell(n)}^{(n)} = 0, e_{h_{\ell(n)}+1, \ell(n)}^{(n)} = \infty, q_{p_{\ell(n)}+1, \ell(n)+1}^{(n)} = \infty,$$

$$e_{p_{\ell(n)}+1, \ell(n)+1}^{(n)} = 0$$

et si $p_{\ell(n)+1} + 2 \leq h_{\ell(n)+1}$, on aura

$$q_{p_{\ell(n)}+2, \ell(n)+1}^{(n)} = -B_{p_{\ell(n)}+2}^{(n)}$$

ALGORITHME qd GENERAL

Les propriétés 2.4 et 2.6 de [1, pages 102 et 121] qui généralisent les relations de récurrence sur deux diagonales adjacentes donnent les quatre relations suivantes :

- Si $P_i^{(n+1)}$ est O.R.

$$P_i^{(n+1)}(x) = \omega_{i-\text{pr}(i+1,n)}^{(n)}(x) P_{\text{pr}(i+1,n)}^{(n)}(x) + E_{i+1}^{(n)} P_{\text{pr}(\text{pr}(i+1,n), n+1)}^{(n+1)}(x)$$

$$Q_i^{(n+1)}(t) = \omega_{i-\text{pr}(i+1,n)}^{(n)}(t) (t Q_{\text{pr}(i+1,n)}^{(n)}(t) - c_n P_{\text{pr}(i+1,n)}^{(n)}(t)) \\ + E_{i+1}^{(n)} Q_{\text{pr}(\text{pr}(i+1,n), n+1)}^{(n+1)}(t)$$

$$\text{avec } E_{i+1}^{(n)} = \begin{cases} -e_i^{(n)} & \text{si } P_{i-1}^{(n+1)} \text{ est O.R.} \\ c_{i+1}^{(n)} & \text{sinon} \end{cases}$$

et si de plus $P_{i+1}^{(n)}$ est O.R. alors

$$P_{i+1}^{(n)}(x) = x P_i^{(n+1)}(x) - q_{i+1,\ell}^{(n)} P_{\text{pr}(i+1,n)}^{(n)}(x)$$

$$Q_{i+1}^{(n)}(t) = Q_i^{(n+1)}(t) - q_{i+1,\ell}^{(n)} Q_{\text{pr}(i+1,n)}^{(n)}(t) + c_n P_i^{(n+1)}(t)$$

On peut calculer ces quatre polynômes à partir des précédents, si on connaît les coefficients $q_{i+1,\ell}^{(n)}$, $E_{i+1}^{(n)}$ et le polynôme $\omega_{i-\text{pr}(i+1,n)}^{(n)}(x)$.

Les coefficients $q_{i+1,\ell}^{(n)}$ et $E_{i+1}^{(n)}$ se calculent à l'aide des relations

de récurrence ayant au plus quatre termes qui ne sont pas toujours aux sommets d'un losange [1, pages 138-168].

Deux cas sont possibles :

I) - Si les cinq polynômes $P_i^{(n)}(x)$, $P_{i+1}^{(n)}(x)$, $P_{i-1}^{(n+1)}(x)$, $P_i^{(n+1)}(x)$ et $P_{i+1}^{(n+1)}(x)$ sont orthogonaux réguliers alors

$$p_{\ell}^{(n)+1} \leq i \leq h_{\ell}^{(n)}$$

$$p_{\ell}^{(n+1)+2} \leq i \leq h_{\ell}^{(n+1)}$$

et nous avons les relations rhomboïdales suivantes :

$$q_{i+1,\ell}^{(n+1)} + e_{i,\ell}^{(n+1)} = q_{i+1,\ell}^{(n)} + e_{i+1,\ell}^{(n)} \quad (1)$$

$$q_{i,\ell}^{(n+1)} e_{i,\ell}^{(n+1)} = q_{i+1,\ell}^{(n)} e_{i,\ell}^{(n)} \quad (2)$$

Pour le démarrage on posera

$$\begin{cases} p_{-1}^{(n)} \text{ O.R. et } p_{-1}^{(n)}(x) = 0 \\ p_0^{(n)} \text{ O.R. et } p_0^{(n)}(x) = 1 \\ e_0^{(n)} = q_0^{(n)} = 0 \end{cases}$$

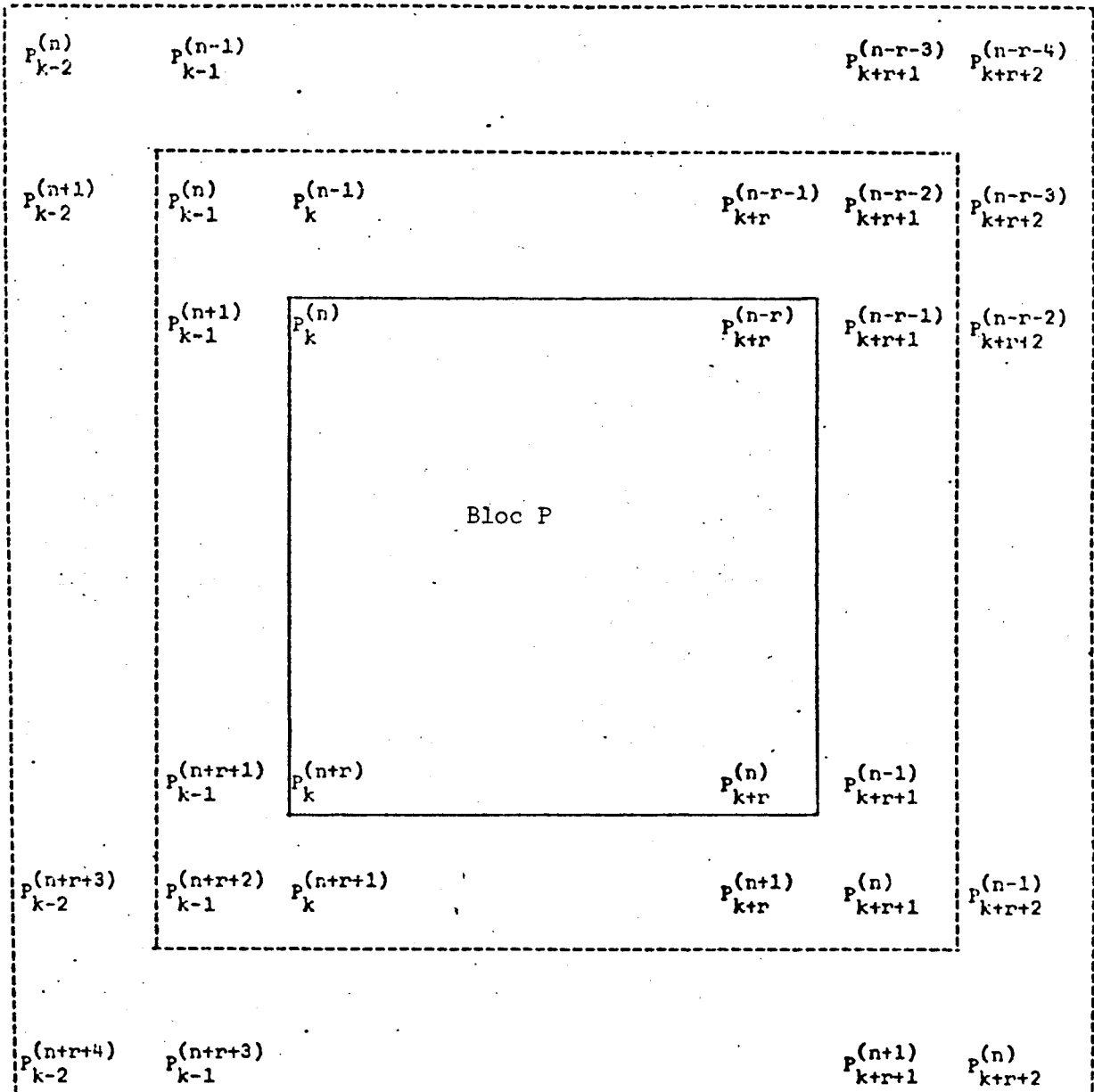
et si $c_n \neq 0$ $p_1^{(n)}$ est O.R. et $q_1^{(n)} = c_{n+1}/c_n$

sinon $p_1^{(n)}$ non O.R., on est en présence d'un bloc,

on est donc dans le cas II).

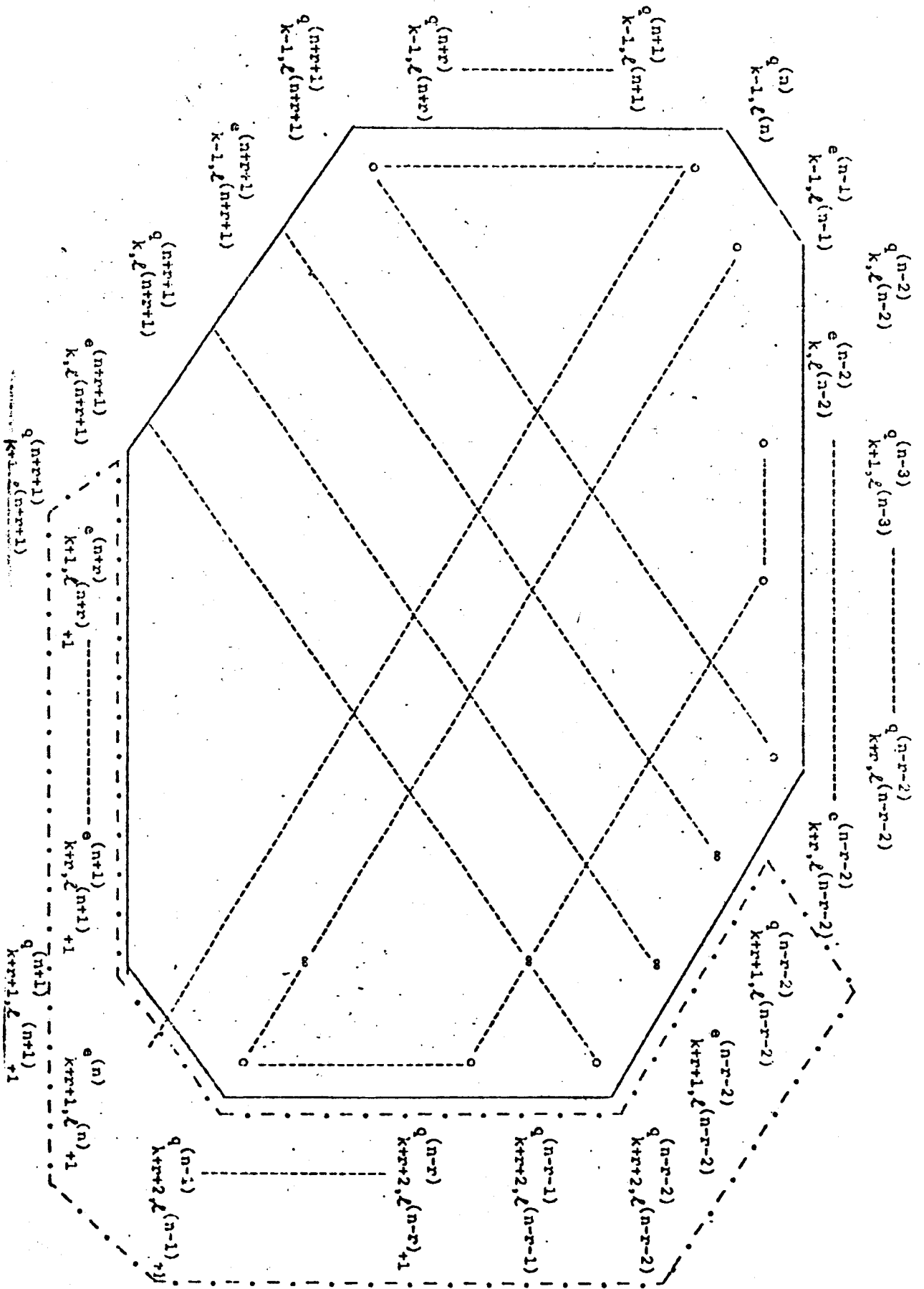
II) - Si l'un au moins des cinq polynômes $p_i^{(n)}(x)$, $p_{i+1}^{(n)}(x)$, $p_{i-1}^{(n+1)}(x)$, $p_i^{(n+1)}(x)$ et $p_{i+1}^{(n+1)}(x)$ est non O.R., on est en présence d'un bloc.

Soient (n, k, r) les coordonnées de ce bloc :



A ce bloc P correspond le bloc qd suivant :





Dans tout ce qui suit les $B_k^{(n)}$ et $C_k^{(n)}$ utilisés sont ceux qui interviennent dans la relation de récurrence à trois termes le long d'une diagonale.

a) Considérons les polynômes $P_{k+l+i}^{(n+r-i)}(x)$ pour $i \in \mathbb{N}$, $0 \leq i \leq r$

1 - Si $P_{k-2}^{(n+r+1-i)}(x)$ est O.R.

$$\begin{aligned} B_{k+l+i}^{(n+r-i)} &= C_{k+i}^{(n+r+1-i)} - q_{k+l+i, \ell}^{(n+r-i)} + 1 \\ C_{k+l+i}^{(n+r-i)} &= C_{k+i}^{(n+r+1-i)} q_{k-1, \ell}^{(n+r-i)} \\ \omega_{i+1}^{(n+r-i)}(x) &= x \omega_i^{(n+r+1-i)}(x) + B_{k+i}^{(n+r+1-i)} \end{aligned}$$

2 - Si $P_{k-2}^{(n+r+1-i)}(x)$ est non O.R.

$$\begin{aligned} B_{k+l+i}^{(n+r-i)} &= -q_{k+l+i, \ell}^{(n+r-i)} + 1 \\ C_{k+l+i}^{(n+r-i)} &= C_{k+i}^{(n+r+1-i)} \\ \omega_{i+1}^{(n+r-i)}(x) &= x \omega_i^{(n+r+1-i)}(x) + B_{k+i}^{(n+r+1-i)} \end{aligned}$$

b) Considérons $P_{k+l+i}^{(n+r+1-i)}(x)$ pour $i \in \mathbb{N}$, $0 \leq i \leq r$

1 - S'il est O.R.

$$\begin{aligned} e_{k+l+i, \ell}^{(n+r-i)} + 1 &= q_{k+l+i, \ell}^{(n+r+1-i)} + e_{k+i, \ell}^{(n+r+1-i)} + \gamma_i \\ q_{k+l+i, \ell}^{(n+r-i)} + 1 &= e_{k+i, \ell}^{(n+r+1-i)} + q_{k+i, \ell}^{(n+r+1-i)} + \gamma_i \end{aligned}$$



2 - S'il n'est pas O.R.

On sait d'après la propriété 2.2. que $q_{k+1+i, \ell}^{(n+r-i)} = 0$ et $e_{k+1+i, \ell}^{(n+r-i)} = \infty$

c) Considérons $P_{k+r+1}^{(n-r-1)}(x)$

1 - Si $P_{k+r}^{(n-r-2)}(x)$ est O.R.

$$E_{k+r+1}^{(n-r-2)} = \begin{cases} -e_{k+r}^{(n-r-2)} & \text{si } P_{k+r-1}^{(n-r-1)}(x) \text{ est O.R.} \\ C_{k+r+1}^{(n-r-2)} & \text{sinon} \end{cases}$$

$$B_{k+r+1}^{(n-r-1)} = -q_{k+r+1, \ell}^{(n-r-2)} - e_{k+r+1, \ell}^{(n-r-2)}$$

$$C_{k+r+1}^{(n-r-1)} = q_{k+r+1, \ell}^{(n-r-2)} E_{k+r+1}^{(n-r-2)}$$

2 - Si $P_{k+r}^{(n-r-2)}(x)$ est quasi orthogonal

$$B_{k+r+1}^{(n-r-1)} = -e_{k+r+1, \ell}^{(n-r-2)}$$

$$C_{k+r+1}^{(n-r-1)} = -q_{k+r+1, \ell}^{(n-r-2)}$$

d) Considérons $P_{k+r+1}^{(n-r-1+i)}(x)$ pour $i \in \mathbb{N}$, $0 \leq i \leq r$

1 - Si $P_{k+r-1-i}^{(n-r-1+i)}(x)$ est O.R.

$$C_{k+r+1}^{(n-r-1+i)} + C_{k+r+2}^{(n-r-1+i)} = B_{k+r+1}^{(n-r+i)}$$

$$-C_{k+r+1}^{(n-r-1+i)} E_{k+r-i}^{(n-r-1+i)} = C_{k+r+1}^{(n-r+i)} \quad (*)$$

$$\omega_{i+1}^{(n-r+i)}(x) = x\omega_i^{(n-r-1+i)}(x) + B_{k+r+1}^{(n-r-1+i)}$$

2 - Si $P_{k+r-1-i}^{(n-r-1+i)}(x)$ est quasi orthogonal

$$C_{k+r+2}^{(n-r-1+i)} = B_{k+r+1}^{(n-r+i)}$$

$$C_{k+r+1}^{(n-r-1+i)} = C_{k+r+1}^{(n-r+i)} \quad (*)$$

$$\omega_{i+1}^{(n-r+i)}(x) = x\omega_i^{(n-r-1+i)}(x) + B_{k+r+1}^{(n-r-1+i)}$$

Remarques : 1/ On démontre les égalités suivantes

$$B_{k+r+1}^{(n-r-1+i)} = B_{k+i}^{(n+r+1-i)}$$

pour $i \in \mathbb{N}$,
 $0 \leq i \leq r$

$$\omega_i^{(n-r-1+i)}(x) = \omega_i^{(n+r+1-i)}(x)$$

à partir de a) et d).



2/ Si $k = 1$, on sait que $C_{k+i}^{(n+r+1-i)} = 0$ pour $0 \leq i \leq r+1$. Pour éviter que tous les $C_{k+r+1}^{n-r-1+i}$ pour $i \in \mathbb{N}$, $0 \leq i \leq r$, soient nuls on démarre les relations (*) à partir de $C_{r+2}^{(n-1)} = -\frac{C_{n+r+1}}{C_{n-1}}$.

e) Considérons $P_{k+r+2}^{(n-r-2+i)}(x)$ pour $i \in \mathbb{N}$, $0 \leq i \leq r+1$

1 - Si $P_{k+r+2}^{(n-r-2+i)}(x)$ est O.R.

On pose

$$E_{k+r+2}^{(n-r-2)} = -e_{k+r+1, (n-r-2)}^{(n-r-2)}$$

$$E_{k+r+2}^{(n-r-2+i)} = C_{k+r+2}^{(n-r-2+i)} \text{ pour } i \in \mathbb{N}, 1 \leq i \leq r+1$$

$$\text{et } \delta_i = \begin{cases} 0 & \text{si } i = 0 \text{ ou } 1 \\ 1 & \text{sinon} \end{cases}$$

1.1. Si $P_{k+r+2}^{(n-r-1+i)}(x)$ est O.R.

$$B_{k+r+2}^{(n-r-1+i)} = -q_{k+r+2, \ell^{(n-r-1+i)} + \gamma_i}^{(n-r-1+i)} = -e_{k+r+2, \ell^{(n-r-2+i)} + \delta_i}^{(n-r-2+i)} - q_{k+r+2, \ell^{(n-r-2+i)} + \delta_i}^{(n-r-2+i)}$$

$$C_{k+r+2}^{(n-r-1+i)} = E_{k+r+2}^{(n-r-2+i)} q_{k+r+2, \ell^{(n-r-2+i)} + \delta_i}^{(n-r-2+i)}$$

1.2. Si $P_{k+r+2}^{(n-r-1+i)}(x)$ est non O.R.

$$q_{k+r+2, \ell^{(n-r-2+i)} + \delta_i}^{(n-r-2+i)} = 0$$

2 - Si $P_{k+r+2}^{(n-r-2+i)}(x)$ n'est pas O.R.

$$e_{k+r+1, \ell}^{(n-r-2+i)} + \delta_i = 0 \text{ avec } \delta_i = \begin{cases} 0 & \text{si } i = 0 \text{ ou } 1 \\ 1 & \text{sinon} \end{cases}$$

f) Considérons $P_{k+r+2}^{(n-1)}(x)$

1 - Si $P_{k+r+2}^{(n-1)}(x)$ est O.R.

1.1. Si $P_{k+r+2}^{(n)}$ est O.R.

$$\begin{aligned} q_{k+r+2, \ell}^{(n)} + e_{k+r+1, \ell}^{(n)} &= e_{k+r+2, \ell}^{(n-1)} + \gamma_r + q_{k+r+2, \ell}^{(n-1)} + \gamma_r \\ - q_{k+r+1, \ell}^{(n)} + e_{k+r+1, \ell}^{(n)} &= c_{k+r+2}^{(n-1)} q_{k+r+2, \ell}^{(n-1)} + \gamma_r \end{aligned}$$

1.2. Si $P_{k+r+2}^{(n)}(x)$ est non O.R.

$$e_{k+r+2, \ell}^{(n-1)} + \gamma_r = \infty$$

2 - Si $P_{k+r+2}^{(n-1)}(x)$ est non O.R.

$$e_{k+r+1, \ell}^{(n-1)} + \gamma_r = 0$$



Remarque 2.3. : Dans l'algorithme que l'on vient de décrire, on ne présente que les relations utilisées dans les sous-programmes qui sont proposés, on pourra consulter [1] pour les autres relations.

Conduite des calculs :

On calcule les éléments de la table qd colonne après colonne. Un bloc est détecté par la présence de zéro dans les colonnes e, le nombre de zéro(s) consécutif(s) donne la largeur du bloc P.

Dans l'algorithme de calcul on ne se limitera pas seulement à se demander si on est en présence d'un bloc car on devra aussi connaître le côté ou le coin pour choisir les relations qui conviennent.

II. ALGORITHME DE CALCUL

A) Par la Méthode normale

La méthode normale consiste à calculer les éléments de la table qd colonne après colonne à partir de deux colonnes d'initialisation : les colonnes e_0 et q_1 .

Description de l'algorithme :

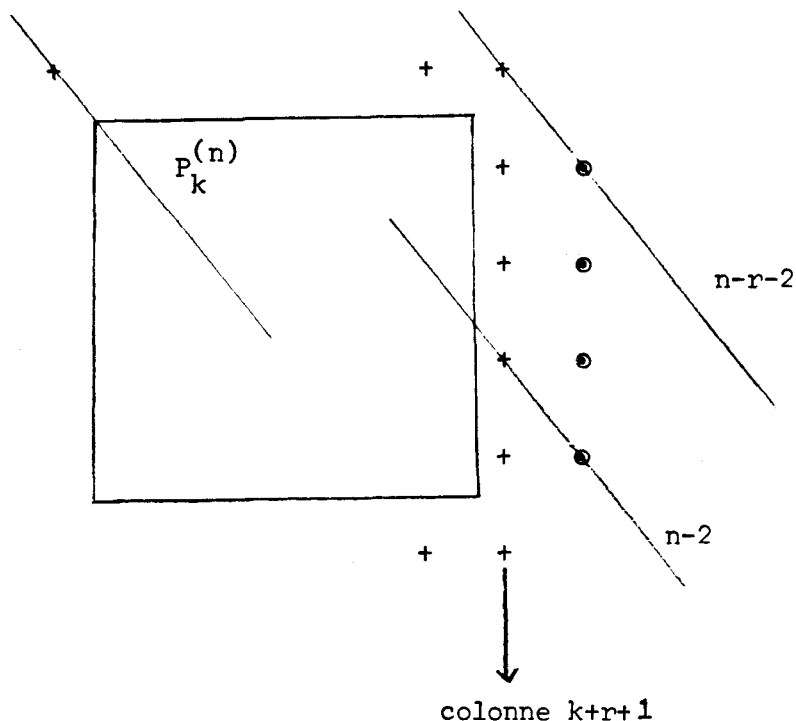
Remarque : On écrira $q_k^{(n)}$ (resp. $e_k^{(n)}$) au lieu de $q_{k,l}^{(n)}$ (resp. $e_{k,l}^{(n)}$) chaque fois que l'on pourra se passer de l'indice $l^{(n)}$.

Pour chaque indice de colonne l , on calcule d'abord les $q_l^{(m)}$ et ensuite les $e_l^{(m)}$, si $e_l^{(m)}$ est nul on est à l'Ouest d'un bloc et le nombre de(s) $e_l^{(m)}$ nul(s) consécutif(s) correspond à la largeur du bloc.

On n'utilise pas les relations rhomboïdales quand on est au Nord, au Sud, sur les deux colonnes de l'Est d'un bloc, ou dans les coins Nord-Est ou Sud-Est d'un bloc, on applique plutôt les relations particulières de l'algorithme qd. Au cours de l'algorithme on se demandera toujours si l'on est dans un cas qui nécessite des relations particulières.

1 - Pour calculer les $q_\ell^{(m)}$ on se pose les questions suivantes en respectant l'ordre

a) Existe-t-il un bloc (n, k, r) tel que $n-r-2 \leq m \leq n-2$ et $\ell = k+r+2$?



o positions que $P_\ell^{(m)}$ peut occuper si le test est positif.

C'est-à-dire si $P_{\ell-2}^{(m+2)}$ est non O.R. et $P_{\ell-1}^{(m+1)}$ est O.R. ou encore, si $P_\ell^{(m)}$ est sur la 2ème colonne après un bloc ?

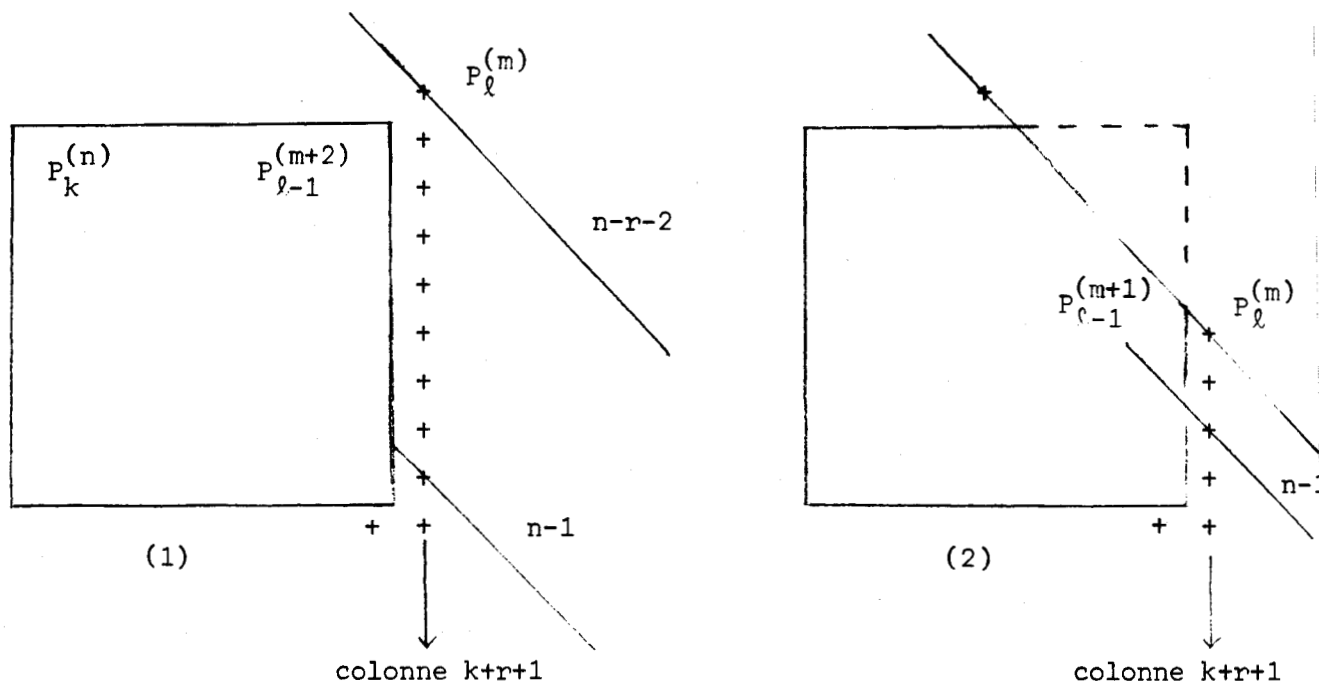
Si oui, ce bloc (n, k, r) est unique et $P_\ell^{(m)}$ peut s'écrire sous la forme $P_{k+r+2}^{(n-r-2+s)}$ avec $s \geq 0$, alors on calcule les $q_{k+r+2}^{(j)}$ pour $j \in \mathbb{N}$, $m \leq j \leq n-2$

Remarque : L'intérêt de ce test est de savoir si l'on est sur la 2ème colonne à l'Est d'un bloc, c'est pour cette raison qu'il ne porte pas sur $P_\ell^{(m)}$. Ce test considère aussi le cas où le bloc P est tronqué.

b) Si $P_\ell^{(m)}$ est non O.R. alors $q_\ell^{(m)}$ est indéfini

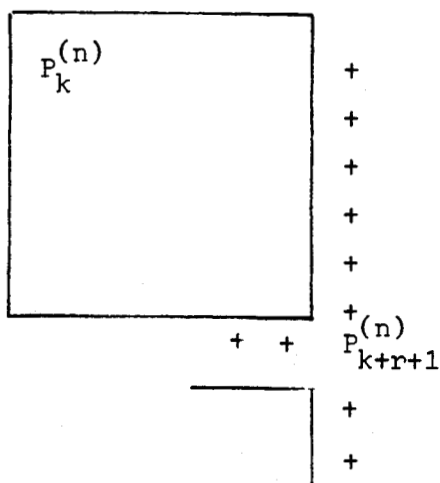
c) Si $P_{\ell}^{(m+1)}$ est non O.R. alors $P_{\ell}^{(m)}$ est au Nord d'un bloc et donc $q_{\ell}^{(m)} = 0$.

d) Existe-t-il un bloc (n, k, r) tel que $n-r-2 \leq m \leq n-1$ et $\ell = k+r+1$?

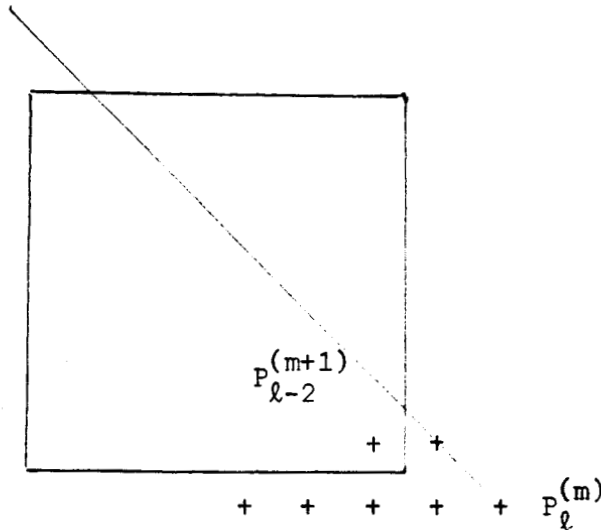


C'est-à-dire si $P_{\ell-1}^{(m+2)}$ (resp. $P_{\ell-1}^{(m+1)}$) est non O.R. dans le cas (1) (resp. (2)) ou encore si $P_{\ell}^{(m)}$ est dans le coin Nord-Est (resp. à l'Est) d'un bloc ?

- Si oui, ce bloc (n, k, r) est unique. On calcule les $c_{k+r+2}^{(j)}$ pour $j \in \mathbb{N}$, $\max(n-r-1, m) \leq j \leq n-1$ et $q_{k+r+1}^{(n-r-2)}$ si on est dans le cas (1). On calcule $c_{k+r+2}^{(n-1)}$ après $q_{k+r+1}^{(n)}$, le calcul de ce dernier est compliqué dans le cas où le polynôme $P_{k+r+1}^{(n)}$ est dans le coin Nord-Est d'un bloc, c'est-à-dire si on a



- e) Si $P_{\ell-1}^{(m)}$ est non O.R. alors $P_{\ell}^{(m)}$ est au Sud ou au Sud-Est d'un bloc (n, k, r) et donc peut s'écrire sous la forme P_{k+1+j}^{n+r-j} avec $j \geq 0$, on applique la relation correspondant à ce cas.
- f) Si $P_{\ell-2}^{(m+1)}$ est non O.R. alors il existe un bloc (n, k, r) et un seul tel que $m = n-1$ et $\ell = k+r+2$, et donc $P_{\ell}^{(m)}$ est de la forme $P_{k+r+2}^{(n-1)}$ avec (n, k, r) les coordonnées d'un bloc.



On calcule $q_{k+r+2}^{(n-1)}$.

- g) Si les réponses aux a), b), c), d), e) et f) sont négatives on utilise la relation rhomboïdale (2) pour calculer $q_{\ell}^{(m)}$.

2 - Considérons $e_{\ell}^{(m)}$

- a) On commence par le même test que pour le calcul de $q_{\ell}^{(m)}$, si on est sur la 2ème colonne après un bloc on calcule les $e_{k+r+2}^{(s)}$ pour $m \leq s \leq n-2$.
- b) Si $P_{\ell}^{(m)}$ ou $P_{\ell}^{(m+1)}$ non O.R. alors $e_{\ell}^{(m)}$ est indéfini.
- c) Si $P_{\ell-1}^{(m+2)}$ (resp. $P_{\ell-1}^{(m+1)}$) est non O.R., alors on est dans le coin Nord-Est (resp. à l'Est) d'un bloc (n, k, r) qui est d'ailleurs unique. Si on est dans le coin Nord-Est on calcule $e_{k+r+1}^{(n-r-2)}$.
- d) Si $P_{\ell-1}^{(m)}$ est non O.R., on est au Sud ou au Sud-Est d'un bloc et donc $P_{\ell}^{(m)}$ peut s'écrire sous la forme $P_{k+1+j}^{(n+r-j)}$ avec $j \geq 0$, on utilise la relation correspondant à ce cas.

e) Si les réponses aux a), b), c) et d) sont négatives on applique la relation rhomboïdale (1).

Remarque 2.4. : L'ordre des tests est important, par exemple pour le calcul de $q_l^{(m)}$ les tests e) ou f) ne doivent passer avant d) ou c). Il est sans doute possible de trouver un autre ordre donnant le résultat.

Algorithme :

Notations : - t est le compteur de blocs,

- col est l'indice de la colonne considérée,

- L est l'indice de la diagonale du dernier terme en q ou e pour la colonne col,

- Lo est l'indice de la première diagonale supérieure de la table qd .

- (n_t, k_t, r_t) correspond aux coordonnées du bloc numéro t.

Remarques : - Dans cet algorithme on pose $e_k^{(j)} = c_{k+1}^{(j)}$ si $P_k^{(j)}$ est à l'Est d'un bloc afin d'utiliser un seul test de détection de bloc(s). (test de nullité des e).

- Le passage d'un numéro au suivant se fait en séquence, sauf s'il existe un ordre aller en.

Soit à calculer les éléments de la table qd pour les c_i avec $i = Lo, \dots, 1$

On pose $t = 0$; $j = Lo$; $col = 1$ et $L = n-1$.

1 - Calcul des $q_1^{(j)}$

1.1. Si $c_j \neq 0$ aller en 1.3.

Sinon on est en présence d'un bloc tronqué dès la première colonne.

$t+1 \rightarrow t$; $r_t = 0$; $k_t = 1$; $n_t = j$.

1.2. Détermination de la largeur du bloc tronqué

Si $c_{j+1} \neq 0$, $j+1 \rightarrow j$ aller en 1.3.

$$r_t + 1 \rightarrow r_t$$

$$j+1 \rightarrow j$$

si $j = n-1$, tous les c_i sont nuls sauf peut-être c_n , fin
aller en 1.2.

1.3. Initialisation de $q_1^{(j)}$

$$q_1^{(j)} = \frac{c_{j+1}}{c_j}, j \geq L \text{ aller en 2}$$

si $c_{j+1} \neq 0$, $j+1 \rightarrow j$ aller en 1.3.

sinon on est en présence d'un bloc démarrant à la première colonne.

$$t+1 \rightarrow t$$

$$r_t = 0$$

$$n_t = j+1$$

$$k_t = 1$$

$$j+2 \rightarrow j$$

1.4. Détermination de la largeur du bloc t démarrant à la première colonne

Si $j \geq n$ aller en 1.5.

si $c_j \neq 0$ aller en 1.3.

$$r_t + 1 \rightarrow r_t$$

$$j+1 \rightarrow j$$

aller en 1.4.

1.5. Si $c_n \neq 0$ aller en 2.

$$r_t + 1 \rightarrow r_t$$

2 - Détermination des polynômes non O.R. appartenant aux blocs détectés dès la première colonne

Si $t = 0$ il n'y a pas de bloc sur la première colonne, aller en 3.

Sinon $P_{1+i+j}^{n_v-i}$ non O.R. pour $i, j = 0, \dots, r_v$ et
 $v = 1, \dots, t$.

3 - Calcul des $e_1^{(j)}$

On pose $j = L_0$ et $L = L-1$.

3.1. Si $P_1^{(j)}$ ou $P_1^{(j+1)}$ non O.R. aller en 3.2.

$$e_1^{(j)} = q_1^{(j+1)} - q_1^{(j)}$$

3.2. Si $j \geq L$ aller en 4.

$j+1 \rightarrow j$
 aller en 3.1.

4 - Détection des blocs démarrant à la colonne $col+1$

On pose $t_1 = t$ et $j = L_0$

4.1. Si $P_{col}^{(j)}$ ou $P_{col}^{(j+1)}$ non O.R. aller en 4.3.

si $e_{col}^{(j)} \neq 0$ aller en 4.3.

sinon apparition d'un (nouveau) bloc

$t+1 \rightarrow t$

$n_t = j$

$r_t = 0$

$k_t = col+1$

$j+1 \rightarrow j$

4.2. Si $j > L$ aller en 5.

Si $P_{col}^{(j)}$ non O.R. aller en 4.3.

Si $e_{col}^{(j)} \neq 0$ aller en 4.3.

$r_t + 1 \rightarrow r_t$

$j+1 \rightarrow j$ aller en 4.2.

4.3. Si $j \geq L$ aller en 5.

$j+1 \rightarrow j$ aller en 4.1.

5 - Détermination des polynômes non O.R. appartenant aux blocs détectés à partir de la colonne col

Si $t_1 \geq t$ il n'y a pas de (nouveau) bloc, aller en 6

$P_{k_{v+j+i}}^{n_v-i}$ non O.R. pour $i, j = 0, \dots, r_v$ et $v = t_1 + 1, \dots, t$.

6 - Calcul des $q_{col}^{(j)}$ ($col \geq 2$)

On pose $L = L-1$, $col = col+1$ et $j = L_0$, si $L < L_0$ fin.

6.1. Si $P_{col-2}^{(j+2)}$ non O.R. et $P_{col-1}^{(j+1)}$ O.R. alors $P_{col}^{(j)}$ est à l'Est d'un bloc sur la deuxième colonne, aller en 6.4.

Si $P_{col}^{(j)}$ non O.R. on sait que $q_{col}^{(j)}$ est indéfini, aller en 6.7.

Si $P_{col}^{(j+1)}$ non O.R. alors $P_{col}^{(j)}$ est au Nord d'un bloc et donc $q_{col}^{(j)} = 0$,

on pose $\bar{j} = j$, aller en 6.6.

Si $P_{col-1}^{(j+2)}$ ou $P_{col-1}^{(j+1)}$ non O.R. alors $P_{col}^{(j)}$ est dans le coin Nord-Est ou à l'Est d'un bloc. On pose $\bar{j} = j$, aller en 6.3.

Si $P_{col-1}^{(j)}$ non O.R. alors $P_{col}^{(j)}$ est au Sud ou dans le coin Sud-Est d'un bloc, aller en 6.2.

Si $P_{col-2}^{(j+1)}$ non O.R. aller en 6.5.

On n'est pas en présence d'un bloc, on applique la relation rhomboïdale

$$q_{col}^{(j)} = \frac{q_{col-1}^{(j+1)} e_{col-1}^{(j+1)}}{e_{col-1}^{(j)}}$$

aller en 6.7.

6.2. $q_{col}^{(j)} = q_{col-1}^{(j+1)} e_{col-1}^{(j+1)}$, aller en 6.6.

En effet $q_{col}^{(j)}$ peut être de la forme $q_{k+r+1}^{(n)}$ et dans ce cas on doit calculer $B_{k+r+1}^{(n)}$ et $C_{k+r+2}^{(n-1)}$.

6.3. Il existe $v \in \{1, \dots, t\}$ tel que $n_v - 1 \geq j \geq n_v - r_v - 2$ et $col = k_v + r_v + 1$

On calcule $C_{k_v+i}^{(n_v+r_v+1-i)}$ pour $i = 0, \dots, r_v+1$
 $C_{k_v+r_v+1}^{(n_v-r_v-1+i)}$ pour $i = 0, \dots, r_v$ et tels que $L \geq n_v - r_v - 1 + i \geq Lo$
 $B_{k_v+r_v+1}^{(n_v-r_v-1+i)}$ pour $i = 0, \dots, r_v$ et tels que $L \geq n_v - r_v - 1 + i \geq Lo$
 $C_{k_v+r_v+2}^{(n_v-r_v-1+i)}$ pour $i = 0, \dots, r_v-1$ et tels que $L \geq n_v - r_v - 1 + i \geq Lo$.
 et $q_{k_v+r_v+1}^{(n_v-r_v-2)}$ et $e_{k_v+r_v+1}^{(n_v-r_v-2)}$ si $n_v - r_v - 2 \geq Lo$.

On pose $j = n_v - 1$, aller en 6.6.

Car si $P_{k_v+r_v+1}^{(n_v-r_v-2)}$ est dans le coin Sud-Est d'un bloc (n, k, r) on doit calculer $B_{k+r+1}^{(n)}$ et $C_{k+r+2}^{(n-1)}$.

6.4. Il existe $v \in \{1, \dots, t\}$ tel que $n_v - r_v - 2 \leq j \leq n_v - 2$ et $col = k_v + r_v + 2$.

6.4.1. Si $j > n_v - r_v - 2$ le bloc est donc tronqué dans le coin Nord-Est, aller en 6.4.2.

Si $P_{col}^{(j)}$ non O.R. alors $q_{col}^{(j)}$ est indéfini. $j+1 \rightarrow j$ et aller en 6.4.2.

Si $P_{col}^{(j+1)}$ non O.R. alors $P_{col}^{(j)}$ est au Nord d'un bloc et donc

$q_{col}^{(j)} = 0$, $j+1 \rightarrow j$ et aller en 6.4.2.

$$q_{col}^{(j)} = -C_{col}^{(j+1)} / e_{col-1}^{(j)}$$

$j+1 \rightarrow j$

On rappelle que j est de la forme $n-r-2$ et col de la forme $k+r+2$

6.4.2. Si $j > L$ aller en 6.7.

Si $j \geq n_v - 2$ aller en 6.7.

Si $P_{col}^{(j)}$ non O.R. alors $q_{col}^{(j)}$ est indéfini. $j+1 \rightarrow j$ et aller en 6.4.2.

Si $P_{col}^{(j+1)}$ non O.R. alors $P_{col}^{(j)}$ est au Nord d'un bloc et donc $q_{col}^{(j)} = 0$,

$j+1 \rightarrow j$ et aller en 6.4.2.

$$q_{col}^{(j)} = C_{col}^{(j+1)} / C_{col}^{(j)}$$

(Ici col est de la forme $k+r+2$ et j de la forme $n-r-1+i$ avec $0 \leq i \leq r-1$)

$j+1 \rightarrow j$ et aller en 6.4.2.

6.5. $P_{col}^{(j)}$ est de la forme $P_{k+r+2}^{(n-1)}$ avec (n, k, r) les coordonnées d'un bloc,

on a donc

$$q_{col}^{(j)} = - \frac{q_{col-1}^{(j+1)} e_{col-1}^{(j+1)}}{C_{col}^{(j)}}$$

aller en 6.7.

6.6. Si $\bar{j} = Lo$ aller en 6.7.

Si $P_{col}^{(\bar{j})}$ est O.R. ou $P_{col}^{(\bar{j}-1)}$ non O.R. alors $P_{col}^{(\bar{j})}$ n'est pas de la forme $P_{k+r+1}^{(n)}$, c'est-à-dire $P_{col}^{(\bar{j})}$ n'est pas dans le coin Sud-Est d'un bloc, aller en 6.7.

Sinon alors il existe $v \in \{1, \dots, t\}$ tel que $j = n_v$ et $col = k_v + r_v + 1$.

On considère différents cas pour calculer $B_{k_v+r_v+1}^{(n_v)}$ et $C_{k_v+r_v+2}^{(n_v-1)}$.

Si $k_v = 1$ ou $P_{k_v-2}^{(n_v-1)}$ non O.R. aller en 6.6.1.

$$B_{k_v+r_v+1}^{(n_v)} = C_{k_v+r_v}^{(n_v+1)} - q_{k_v+r_v+1}^{(n_v)}, \text{ aller en 6.6.2.}$$

$$6.6.1. \quad B_{k_v+r_v+1}^{(n_v)} = -q_{k_v+r_v+1}^{(n_v)}$$

6.6.2. Si $P_{k_v-1}^{(n_v-1)}$ non O.R. alors $C_{k_v+r_v+2}^{(n_v-1)} = B_{k_v+r_v+1}^{(n_v)}$, aller en 6.7.

$$\text{Sinon } C_{k_v+r_v+2}^{(n_v-1)} = B_{k_v+r_v+1}^{(n_v)} - C_{k_v+r_v+1}^{(n_v-1)}$$

6.7. Si $j \geq L$ aller en 7.

$j+1 \rightarrow j$ et aller en 6.1.

7 - Calcul des $e_{col}^{(j)}$ ($col \geq 2$)

On pose $L = L-1$ et $j = Lo$, si $L < Lo$ fin

7.1. Si $P_{col-2}^{(j+2)}$ non O.R. et $P_{col-1}^{(j+1)}$ O.R. alors $P_{col}^{(j)}$ est à l'Est d'un bloc sur la 2ème colonne, aller en 7.4.

Si $P_{col}^{(j)}$ ou $P_{col}^{(j+1)}$ non O.R., c'est-à-dire si $P_{col}^{(j)}$ est au Nord d'un bloc ou dans un bloc, aller en 7.5.

Si $P_{col-1}^{(j+1)}$ ou $P_{col-1}^{(j+2)}$ non O.R. alors $P_{col}^{(j)}$ est à l'Est ou dans le coin Nord-Est d'un bloc, aller en 7.3.

Si $P_{col-1}^{(j)}$ non O.R. alors $P_{col}^{(j)}$ est au Sud ou au Sud-Est d'un bloc, aller en 7.2.

On n'est pas en présence d'un bloc, on utilise la relation rhomboïdale

$$e_{col}^{(j)} = q_{col}^{(j+1)} + e_{col-1}^{(j+1)} - q_{col}^{(j)}$$

aller en 7.5.

7.2. $e_{col}^{(j)} = q_{col}^{(j+1)} + e_{col-1}^{(j+1)}$, aller en 7.5.

7.3. Il existe $v \in \{1, \dots, t\}$ tel que $n_v - r_v - 2 \leq j \leq n_v - 1$ et $col = k_v + r_v + 1$.

Tous les termes $e_{k_v + r_v + 1}^{(n_v - r_v - 2)}$ et $C_{k_v + r_v + 2}^{(n_v - r_v - 1 + s)}$ pour $s \in \mathbb{N}$, $0 \leq s \leq r_v$, qui

permettent de continuer les calculs après le bloc (n_v, k_v, r_v) sont déjà connus puisqu'ils ont été calculés en 6.3. et 6.6.

On pose $j = n_v - 1$, aller en 7.5.

7.4. Il existe $v \in \{1, \dots, t\}$ tel que $n_v - r_v - 2 \leq j \leq n_v - 2$ et $col = k_v + r_v + 2$.

7.4.1. Si $j > L$ ou $j > n_v - 2$, $j-1 \rightarrow j$ aller en 7.5.

Si $P_{col}^{(j)}$ ou $P_{col}^{(j+1)}$ non O.R. alors $P_{col}^{(j)}$ est dans un bloc ou au Nord d'un bloc et donc $e_{col}^{(j)}$ est indéfini, $j+1 \rightarrow j$, aller en 7.4.1.

Sinon $e_{col}^{(j)} = q_{col}^{(j+1)} - q_{col}^{(j)}$, $j+1 \rightarrow j$, aller en 7.4.1.

7.5. Si $j \geq L$ aller en 4.

$j+1 \rightarrow j$ et aller en 7.1.

Remarque : Cet algorithme calcule aussi $B_{k+r+1}^{(n)}$ et $C_{k+r+2}^{(n-1)}$ suivant les cas si l'on est en présence d'un bloc (n, k, r) .

On complète l'algorithme précédent avec le calcul des $c_{k+r+2}^{(n-r-1+i)}$ pour $i = 0, \dots, r-1$, $q_{k+r+1}^{(n-r-2)}$ et $e_{k+r+1}^{(n-r-2)}$ qui permettent la poursuite des calculs après un bloc (n, k, r) .

Algorithme de calcul des $c_{k+r+2}^{(n-r-1+i)}$ pour $i = 0, \dots, r-1$, $q_{k+r+1}^{(n-r-2)}$ et $e_{k+r+1}^{(n-r-2)}$:

On considère le cas où l'on doit calculer tous ces coefficients. Dans la mise en oeuvre tous les cas sont considérés, c'est-à-dire même quand les blocs sont tronqués.

Le passage d'un numéro au suivant se fait en séquence sauf s'il existe un ordre aller en.

1 - Calcul des $B_{k+r+1}^{(n-r-1+i)}$ et $c_{k+i}^{(n+r+1-i)}$ pour $i = 0, \dots, r$

Si $k \geq 2$ aller en 1.2.

Cas où le bloc commence à la première colonne

On sait que $c_{k+i}^{(n+r+1-i)} = 0$ pour $i = 0, \dots, r$

et $B_{k+r+1}^{(n-r-1+i)} = -q_{k+i}^{(n+r+1-i)}$ pour $i = 0, \dots, r$

aller en 2.

1.2. Cas où le bloc ne commence pas sur la première colonne

1.2.1. Si $P_{k-2}^{(n+r+2)}$ est non O.R., c'est-à-dire si $P_{k-1}^{(n+r+1)}$ est à l'Est d'un bloc, aller en 1.2.2.

Sinon $c_k^{(n+r+1)} = -q_{k-1}^{(n+r+1)} e_{k-1}^{(n+r+1)}$

et $B_{k+r+1}^{(n-r-1)} = -q_k^{(n+r+1)} - e_{k-1}^{(n+r+1)}$

aller en 1.2.3.

$$1.2.2. \quad B_{k+r+1}^{(n-r-1)} = -q_k^{(n+r+1)}$$

$C_k^{(n+r+1)}$ est déjà connu puisque $P_{k-1}^{(n+r+1)}$ est à l'Est d'un bloc.

1.2.3. On pose $j = 1$

1.2.3.1. Si $P_{k-2}^{n+r+2-j}$ est non O.R., c'est-à-dire si le polynôme $P_{k-1}^{n+r+1-j}$ est à l'Est d'un bloc, alors aller en 1.2.3.2.

$$\text{Sinon } C_{k+j}^{(n+r+1-j)} = C_{k+j-1}^{(n+r+2-j)} \times q_{k-1}^{(n+r+1-j)}$$

$$\text{et } B_{k+r+1}^{(n-r-1+j)} = C_{k+j-1}^{(n+r+2-j)} - q_{k+j}^{(n+r+1-j)}$$

aller en 1.2.3.3.

$$1.2.3.2. \quad C_{k+j}^{(n+r+1-j)} = C_{k+j-1}^{(n+r+2-j)}$$

$$\text{et } B_{k+r+1}^{(n-r-1+j)} = -q_{k+j}^{(n+r+1-j)}$$

1.2.3.3. Si $j \geq r$ aller en 2

$j+1 \rightarrow j$ et aller en 1.2.3.1.

2 - Calcul des $C_{k+r+1}^{(n-r-1+i)}$ pour $i = 0, \dots, r+1$

Si $k \geq 2$ aller en 2.1.

On sait que $C_{k+r+1}^{(n)} = 0$ et $C_{k+r+1}^{(n-1)} = -c_{n+r+1}/c_{n-1}$

$r-1 \rightarrow j$, aller en 2.3.

2.1. Si $P_{k-2}^{(n+1)}$ est non O.R., c'est-à-dire si $P_{k-1}^{(n)}$ est à l'Est d'un bloc,
 aller en 2.2.

$$\text{Sinon } C_{k+r+1}^{(n)} = C_{k+r}^{(n+1)} \times q_{k-1}^{(n)}$$

$r \rightarrow j$ et aller en 2.3.

$$2.2. \underline{C_{k+r+1}^{(n)} = C_{k+r}^{(n+1)}}$$

$r \rightarrow j$

2.3. Si $j < 0$, aller en 3.

Si $P_{k+r-1-j}^{(n-r-1+j)}$ est non O.R., c'est-à-dire si $P_{k+r-j}^{(n-r-1+j)}$ est au Sud-est ou au Sud d'un bloc, aller en 2.4.

$$C_{k+r+1}^{(n-r-1+j)} = -C_{k+r+1}^{(n-r+j)} / E_{k+r-j}^{(n-r-1+j)}$$

$$\text{avec } E_{k+r-j}^{(n-r-1+j)} = \begin{cases} -e_{k+r-1-j}^{(n-r-1+j)} & \text{si } P_{k+r-2-j}^{(n-r+j)} \text{ est O.R.} \\ C_{k+r-j}^{(n-r-1+j)} & \text{sinon} \end{cases}$$

aller en 2.5.

$$2.4. \underline{C_{k+r+1}^{(n-r-1+j)} = C_{k+r+1}^{(n-r+j)}}$$

2.5. $j-1 \rightarrow j$ et aller en 2.3.

3 - Calcul de $q_{k+r+1}^{(n-r-2)}$ et $e_{k+r+1}^{(n-r-2)}$

3.1. Si $P_{k+r}^{(n-r-2)}$ est non O.R., c'est-à-dire si $P_{k+r+1}^{(n-r-2)}$ est au Sud-Est ou au Sud d'un bloc, aller en 3.2.

$$\text{Sinon } q_{k+r+1}^{(n-r-2)} = C_{k+r+1}^{(n-r-1)} / E_{k+r+1}^{(n-r-2)}$$

$$\text{et } e_{k+r+1}^{(n-r-2)} = -B_{k+r+1}^{(n-r-1)} - q_{k+r+1}^{(n-r-2)}$$

$$\text{avec } E_{k+r+1}^{(n-r-2)} = \begin{cases} -e_{k+r}^{(n-r-2)} & \text{si } P_{k+r-1}^{(n-r-1)} \text{ est O.R.} \\ C_{k+r+1}^{(n-r-2)} & \text{sinon} \end{cases}$$

aller en 4.

$$3.2. \underline{\text{Alors } q_{k+r+1}^{(n-r-2)} = -C_{k+r+1}^{(n-r-1)} \text{ et } e_{k+r+1}^{(n-r-2)} = -B_{k+r+1}^{(n-r-1)}}.$$

4 - Calcul des $C_{k+r+2}^{n-r-1+i}$ pour $i = 0, \dots, r-1$

On pose $j = 0$

4.1. Si $j \geq r$ fin

4.2. Si $P_{k+r-1-j}^{(n-r-1+j)}$ est non O.R., c'est-à-dire si $P_{k+r-j}^{(n-r-1+j)}$ est au Sud ou au

Sud-Est d'un bloc, aller en 4.3.

$$\text{Sinon } C_{k+r+2}^{(n-r-1+j)} = B_{k+r+1}^{(n-r+j)} - C_{k+r+1}^{(n-r-1+j)}$$

$j+1 \rightarrow j$, aller en 4.1.

4.3. $C_{k+r+2}^{(n-r-1+j)} = B_{k+r+1}^{(n-r+j)}$

$j+1 \rightarrow j$, aller en 4.1.

B) Par la Méthode progressive

La méthode progressive consiste à calculer la table qd horizontale par horizontale à partir d'une colonne et deux horizontales d'initialisation (la colonne e_0 , la première horizontale est nulle et la seconde vaut un rapport de coefficients).

DRAUX propose dans son livre [1, page 171] un algorithme qui donne la même table que celle calculée par la méthode progressive en utilisant les coefficients de la série réciproque, nous proposons la mise en oeuvre de cet algorithme.

Définition 2.5. : Soient une série formelle $f(x) = \sum_{i=0}^{\infty} c_i x^i$ et s le plus

petit entier tel que $c_s \neq 0$. On appelle série réciproque de f la série

$$f^{-1}(x) = \sum_{i=0}^{\infty} \overline{c}_i x^i \text{ telle que } x^{-s} f(x) f^{-1}(x) = 1. \quad (1)$$

La relation (1) donne le système régulier (2)

$$(2) \quad \begin{pmatrix} & & & c_s \\ & 0 & & \\ & & \ddots & \\ c_s & & & c_{s+k} \end{pmatrix} \begin{pmatrix} c_k \\ \vdots \\ \overline{c_0} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \forall k \geq 0$$

et donc les $\overline{c_i}$ se calculent sans aucune difficulté.

Soient la série formelle $f(x) = \sum_{i=0}^{\infty} c_i x^i$ et $f^{-1}(x) = \sum_{i=0}^{\infty} \overline{c_i} x^i$ sa

série réciproque. On peut définir une fonctionnelle linéaire $\overline{c}$ telle que

$$\overline{c}(x^j) = \overline{c_j} \quad \forall j \in \mathbb{N}$$

Définition 2.6. : On appelle ensemble des polynômes orthogonaux réciproques l'ensemble des polynômes $\{R_k^{(n)}\}$ orthogonaux par rapport aux fonctionnelles linéaires définies par

$$\overline{c}^{(n)}(x^j) = \overline{c}_{n+j} \quad \forall j, n \in \mathbb{N}.$$

On range ces polynômes dans une table à double entrée que l'on appellera table des polynômes orthogonaux réciproques.

A partir de la série réciproque on peut définir les fonctionnelles linéaires $d^{(n)}$ telles que $\forall n \in \mathbb{Z}, \forall j \in \mathbb{N}$

$$d^{(n)}(x^j) = \begin{cases} 0 & \text{si } n+j < 0 \\ \overline{c}_{n+j} & \text{sinon} \end{cases}$$

$\mathbb{Z}$ étant l'ensemble des entiers relatifs.

Définition 2.7. : On appellera table complétée des polynômes orthogonaux réciproques la table des polynômes orthogonaux par rapport aux fonctionnelles linéaires $d^{(n)}$. La table qd de ces polynômes sera appelée table qd complétée.

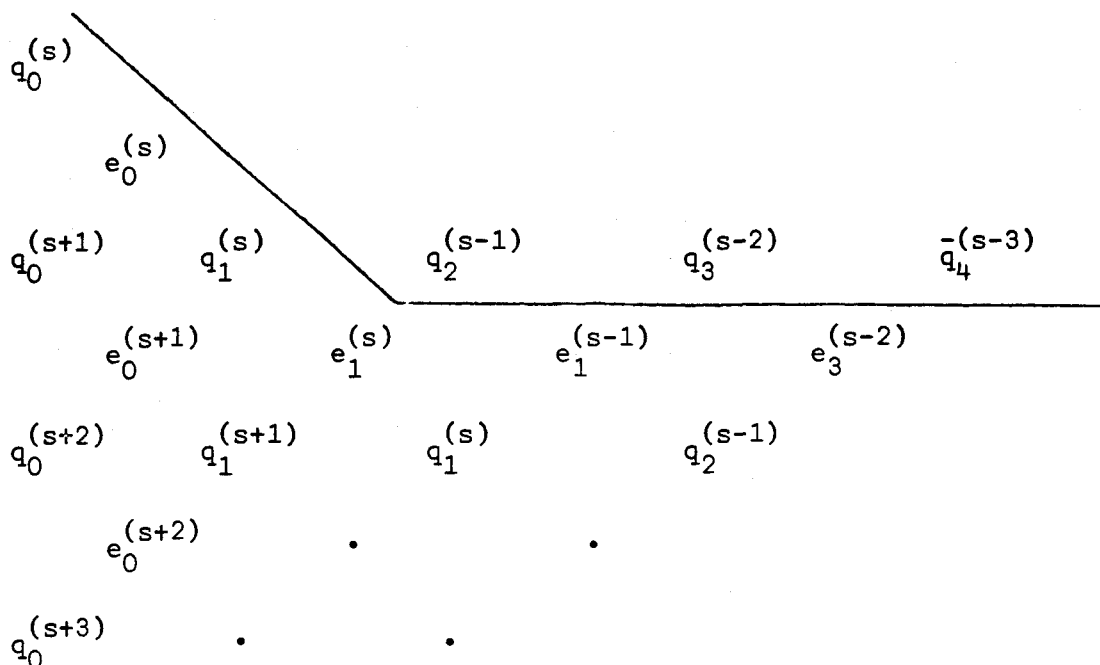
Dans toute la suite les d_i sont les moments des fonctionnelles linéaires $d^{(n)}$.

La table

$\bar{q}_0^{(0)}$				
$\bar{e}_0^{(0)}$				
$\bar{q}_0^{(1)}$	$\bar{q}_1^{(0)}$	$\bar{q}_2^{(-1)}$	$\bar{q}_3^{(-2)}$	$\bar{q}_4^{(-3)}$
$\bar{e}_0^{(1)}$	$\bar{e}_1^{(0)}$	$\bar{e}_2^{(-1)}$	$\bar{e}_3^{(-2)}$	
$\bar{q}_0^{(2)}$	$\bar{q}_1^{(1)}$	$\bar{q}_2^{(0)}$	$\bar{q}_3^{(-1)}$	$\bar{q}_4^{(-2)}$
$\bar{e}_0^{(2)}$	$\bar{e}_1^{(1)}$	$\bar{e}_2^{(0)}$	.	
$\bar{q}_0^{(3)}$	$\bar{q}_1^{(2)}$	.	.	.

→ diagonale 1/2

obtenue en appliquant la méthode normale aux (d_i) est la table qd complétée des polynômes orthogonaux réciproques $\{R_k^{(n)}\}$. De cette table on déduit la table qd des polynômes orthogonaux $\{P_k^{(n)}\}$ par la méthode progressive :



il suffit d'effectuer une symétrie par rapport à la diagonale 1/2 en changeant convenablement $\bar{q}$ en e et $\bar{e}$ en q . C'est-à-dire en utilisant les relations

$$\bar{q}_1^{(0)} = -q_1^{(s)}$$

$$\bar{q}_m^{(n)} = e_{m+n-1}^{(s+1-n)} \text{ et } \bar{e}_m^{(n)} = q_{m+n}^{(s+1-n)} \text{ pour } n > -m \quad (1)$$

$$q_m^{(n+s)} = \bar{e}_{n+m-1}^{(1-n)} \text{ et } e_m^{(n+s)} = \bar{q}_{n+m}^{(1-n)} \text{ pour } n > -m \quad (2)$$

partout où ces valeurs sont définies.

Algorithme de changement de $\bar{q}$ en e et $\bar{e}$ en q

Cet algorithme consiste à chercher les éléments $\bar{q}$ et $\bar{e}$ qui sont définis afin d'appliquer (1) ou (2).

Remarque : Si $R_{k-1}^{(n)}$ est O.R. alors

$$\bar{E}_k^{(n)} = \begin{cases} -\bar{e}_{k-1}^{(n)} & \text{si } R_{k-2}^{(n+1)} \text{ est O.R.} \\ \bar{C}_k^{(n)} & \text{sinon.} \end{cases}$$

Dans notre algorithme nous posons $\bar{e}_{k-1}^{(n)} = \bar{C}_k^{(n)}$ si $R_{k-1}^{(n)}$ est à l'Est d'un bloc, dans ce cas les relations (1) et (2) pourront être appliquées si $\bar{e}$ ou $\bar{q}$ est fini.

Comme dans les autres algorithmes le passage d'un numéro au suivant se fait en séquence sauf s'il y a un ordre aller en.

1 - On pose $L = n-s-1$, $m = 1$, $j = 1$ et $q_1^{(s)} = -\bar{c}_1/\bar{c}_0$

2 - Considérons les relations (2)

2.1. Si $R_{m+j-1}^{(1-j)}$ est non O.R., c'est-à-dire si $R_{m+j-1}^{(1-j)}$ est dans un bloc, on sait

 que $\bar{e}_{m+j-1}^{(1-j)}$ n'est pas fini, aller en 2.2.

Sinon

- Si $R_{m+j-1}^{(2-j)}$ est non O.R., c'est-à-dire si $R_{m+j-1}^{(1-j)}$ est au Nord d'un bloc, on sait que $\bar{e}_{m+j-1}^{(1-j)}$ n'est pas fini, aller en 2.3.

sinon alors $R_{m+j-1}^{(1-j)}$ n'est ni dans un bloc, ni au Nord d'un bloc et donc $\bar{e}_{m+j-1}^{(1-j)}$ est fini. On a

$$P_m^{(s+j)} \text{ O.R. et } q_m^{(s+j)} = \bar{e}_{m+j-1}^{(1-j)}$$

- Si $R_{m+j}^{(1-j)}$ est non O.R., c'est-à-dire si $R_{m+j-1}^{(1-j)}$ est au Nord-Ouest ou à l'Ouest d'un bloc, alors $\bar{q}_{m+j}^{(1-j)}$ n'est pas fini, aller en 2.4.

sinon $\bar{q}_{j+m}^{(1-j)}$ est fini et

$$e_m^{(s+j)} = \bar{q}_{m+j}^{(1-j)}, \text{ aller en 2.7.}$$

2.2. Si $R_{m+j}^{(1-j)}$ ou $R_{m+j-1}^{(2-j)}$ non O.R., c'est-à-dire si $R_{m+j}^{(1-j)}$ est dans un bloc

 ou à l'Est d'un bloc, on sait que $\bar{q}_{m+j}^{(1-j)}$ est indéfini ou infini, aller

 en 2.4.

Sinon $P_m^{(s+j)}$ est O.R. et $e_m^{(s+j)} = \bar{q}_{m+j}^{(1-j)}$

aller en 2.4.

2.3. Alors $P_m^{(s+j)}$ est non O.R.

2.4. Si $j > L$ aller en 2.5.

$j+1 \rightarrow j$ et aller en 2.1.

2.5. Si $R_{m+j}^{(-j)}$ ou $R_{m+j}^{(1-j)}$ est non O.R., c'est-à-dire si $R_{m+j}^{(-j)}$ est au Nord d'un bloc ou dans un bloc, on sait que $\bar{e}_{m+j}^{(-j)}$ n'est pas fini, aller en 2.6.

Sinon $R_{m+j}^{(-j)}$ est O.R. et n'est pas au Nord d'un bloc et donc $\bar{e}_{m+j}^{(-j)}$ est fini. On pose

$$P_m^{(s+j+1)} \text{ O.R. et } q_m^{(s+j+1)} = \bar{e}_{m+j}^{(-j)}, \text{ aller en 2.7.}$$

2.6. Alors $P_m^{(s+j+1)}$ est non O.R.

2.7. Si $L \leq 1$ aller en 3.

$L-1 \rightarrow L, m+1 \rightarrow m, j = 1$ et aller en 2.1.

3 - Considérons les relations (1)

Détermination des éléments $e_j^{(s+1-j)}$

On pose $L = n-s-1$, et $j = 1$

3.1. On pose $P_j^{(s+n-j)}$ O.R.

Si $R_1^{(j)}$ est non O.R. alors $\bar{q}_1^{(j)}$ est indéfini, aller en 3.2.

Sinon $R_1^{(j)}$ est O.R. et comme il n'est pas à l'Est d'un bloc $\bar{q}_1^{(j)}$ est fini. On pose

$$e_j^{(s+1-j)} = \bar{q}_1^{(j)}$$

3.2. Si $j \geq L$ aller en 3.3.

$j+1 \rightarrow j$ et aller en 3.1.

Détermination des autres éléments

On pose $L = L-1$, $m = 1$ et $j = 1$

3.3. Si $R_m^{(j)}$ est non O.R. donc $\bar{e}_m^{(j)}$ est indéfini, aller en 3.4.

Sinon

- Si $R_m^{(j+1)}$ est non O.R. alors $R_m^{(j)}$ est au Nord d'un bloc et donc $\bar{e}_m^{(j)}$ est indéfini, aller en 3.5.

sinon $R_m^{(j)}$ est O.R. et n'est pas au Nord d'un bloc et donc $\bar{e}_m^{(j)}$ est fini. On pose

$$p_{m+j}^{(s+1-j)} \text{ O.R. et } q_{m+j}^{(s+1-j)} = \bar{e}_m^{(j)}$$

- Si $R_{m+1}^{(j)}$ est non O.R. alors $q_{m+1}^{(j)}$ est indéfini, aller en 3.6.

sinon $R_{m+1}^{(j)}$ est O.R. et comme $R_m^{(j+1)}$ est O.R. $R_{m+1}^{(j)}$ n'est pas

à l'Est d'un bloc et donc $\bar{q}_{m+1}^{(j)}$ est fini. On pose

$$e_{m+j}^{(s+1-j)} = \bar{q}_{m+1}^{(j)}, \text{ aller en 3.6.}$$

3.4. Si $R_{m+1}^{(j)}$ ou $R_m^{(j+1)}$ non O.R. alors $R_{m+1}^{(j)}$ est dans un bloc ou à l'Est d'un bloc et donc $\bar{q}_{m+1}^{(j)}$ est indéfini ou infini, aller en 3.5.

Sinon on pose $p_{m+j}^{(s+1-j)} \text{ O.R. et } e_{m+j}^{(s+1-j)} = \bar{q}_{m+1}^{(j)}$, aller en 3.6.

3.5. Alors $p_{m+j}^{(s+1-j)}$ est non O.R.

3.6. Si $j \geq L$ aller en 3.7.

$j+1 \rightarrow j$ et aller en 3.3.

3.7. Si $L \leq 1$ fin

$L-1 \rightarrow L$, $m+1 \rightarrow m$, $j = 1$ et aller en 3.3.

Remarque 2.8. : La table qd obtenue par la méthode progressive de [1] est la table qd complétée des polynômes orthogonaux. Elle peut être obtenue directement par la méthode normale, il suffit d'introduire un bloc infini, c'est-à-dire ajouter des coefficients $c_i = 0$ pour $i < 0$ et, calculer la table qd par la méthode normale en choisissant convenablement l'indice de la première diagonale supérieure.

III. MISE EN OEUVRE

La mise en oeuvre se fait à l'aide de trois sous-programmes : QUDE, DETECT et EST.

Dans les sous-programmes :

- Les coefficients c_i de la série formelle sont notés $DC(I)$.
- Les éléments $q_k^{(j)}$ (resp. $e_k^{(j)}$) de la table qd sont notés $DQ(J, K)$ (resp. $DE(J, K)$).
- Le tableau NDF , à double entrée, qui permet de connaître l'allure de la table P des polynômes orthogonaux est tel que

$$NDF(J, K) = 0 \text{ si le polynôme } P_k^{(j)} \text{ est non O.R.} \\ = 1 \text{ sinon}$$

Les résultats sont échangés entre le programme principal et les sous-programmes par l'intermédiaire de l'instruction COMMON, il s'agit des tableaux DC , DQ , DE , NDF et, DBR et DCR qui sont les tableaux des $B_k^{(j)}$ et $C_k^{(j)}$.

Dans les sous-programmes QUDE et EST on posera $e_k^{(j)} = C_{k+1}^{(j)}$ si le polynôme $P_k^{(j)}$ est à l'Est d'un bloc pour des raisons évoquées dans l'algorithme de calcul des éléments de la table qd .

Mode d'emploi des sous-programmes

* * * QUDE * * * *

Appel CALL QUDE(NL, MOI, MA, BLOC)

Paramètres : - Le tableau des coefficients de la série formelle est indicé de MOI à NL.

- MA détermine le choix de(s) méthode(s) :

- si MA = 0 méthode normale
MOI peut être égal à zéro ou à - NL, il correspond à l'indice de la première diagonale supérieure de la table qd.
- si MA < 0 méthode progressive. Dans ce cas MOI = 0.
- si MA > 0 méthodes normale et progressive. MOI doit être égal à zéro ou à - NL, car il est choisi égal à zéro dans la méthode progressive.

- BLOC est le nombre de bloc(s) dans la table qd.

- MAT est le numéro logique de l'imprimante.

NL, MOI et MA sont des paramètres d'entrée et BLOC est un paramètre de sortie.

Remarques : 1° La propagation des erreurs de calcul étant importante la valeur de DEPS qui intervient dans le test de nullité des e, c'est-à-dire dans la détection des blocs, peut être choisie égale à 10^{-10} .

2° Les blocs sont affectés des numéros correspondant à l'ordre de détection.

Sous-programmes utilisés : DETECT et EST.

Description : Calcul de la table qd des polynômes orthogonaux par la méthode normale ou par la méthode progressive suivant la valeur de MA. Ce programme donne aussi l'allure de la table P des polynômes orthogonaux.

* * * EST * * * *

Appel : CALL EST(N, NR, K, MON, MAX)

Paramètres : - (N, NR, K) sont les coordonnées du bloc.

- MON est l'indice de la première diagonale supérieure de la table qd et MAX celui de la dernière diagonale pour la colonne q d'indice $K+NR+1$.

Tous les paramètres sont des paramètres d'entrée.

Description : Calcul des éléments $c_{k+r+2}^{(n-r-1+i)}$ et $q_{k+r+1}^{(n-r-2)}$ et $e_{k+r+1}^{(n-r-2)}$ selon les cas, qui permettent de continuer le calcul des éléments de la table qd après le bloc (n, k, r).

* * * DETECT * * * *

Appel : CALL DETECT(J, MET, I, LA, L)

Paramètres : J est l'indice de la diagonale à partir de laquelle on constate que l'on est dans le coin Nord-Est ou à l'Est (sur la première colonne ou la deuxième) d'un bloc.

I est l'indice de la colonne dont on calcule les éléments qd.

MET est un entier qui vaut 1 si l'on est dans le coin Nord-Est ou sur la première à l'Est d'un bloc, et 2 si l'on est sur la deuxième.

LA est le nombre de bloc(s) déjà détecté(s).

L est le numéro du bloc que l'on contourne.

J, MET, I et LA sont des paramètres d'entrée et L le paramètre de sortie.

Description : Ce sous-programme détermine le bloc que l'on contourne en donnant son numéro L.

4 - Exemples d'appel des sous-programmes

Le programme ci-dessous lit les valeurs de NL, de MON, de MAX et des coefficients c_i , et appelle le sous-programme QUDE, ce dernier utilise les sous-programmes DETECT et EST.

```

1. *****
2. C PROGRAMME PRINCIPAL POUR LE CALCUL DE LA TABLE 0-0 PAR LES METHODES
3. C NORMALE OU PROGRESSIVE
4. C *****
5. C ***** CARTES SUSCEPTIBLES D'ETRE MODIFIEES PAR L'UTILISATEUR *****
6. COMMON/DCK/(-20:20,J:20),DC(-20:20),DBR(-20:20,0:20),DC(-20:20,0:20)
7. 1,DE(-20:20,0:20),NDF(-20:20,0:20)/S/N(20),K(20),NR(20)
8. LGB=105
9. C ***** FIN *****
10. READ(LGB,10)NL
11. READ(LGB,10)MON
12. READ(LGB,10)MAX
13. READ(LGB,20)(DC(I),I=MON,NL)
14. 10 FORMAT(I5)
15. 20 FORMAT(5F12.0)
16. CALL QUDE(NL,MON,MAX,LA):STOP
17. END

```

Essais numériques

1° Pour NL = 10, MON = -10, MAX = 0 (méthode normale) et les coefficients c_i suivants :

```

C(-10)= .0000000000000000+00
C(-9)= .0000000000000000+00
C(-8)= .0000000000000000+00
C(-7)= .0000000000000000+00
C(-6)= .0000000000000000+00
C(-5)= .0000000000000000+00
C(-4)= .0000000000000000+00
C(-3)= .0000000000000000+00
C(-2)= .0000000000000000+00
C(-1)= .0000000000000000+00
C(0)= .1000000000000000+02
C(1)= .9000000000000000+01
C(2)= .8000000000000000+01
C(3)= .7000000000000000+01
C(4)= .6000000000000000+01
C(5)= -.6000000000000000+01
C(6)= .6000000000000000+01
C(7)= -.6000000000000000+01
C(8)= .5000000000000000+01
C(9)= .4000000000000000+01
C(10)= .3000000000000000+01

```

On obtient les résultats suivants :

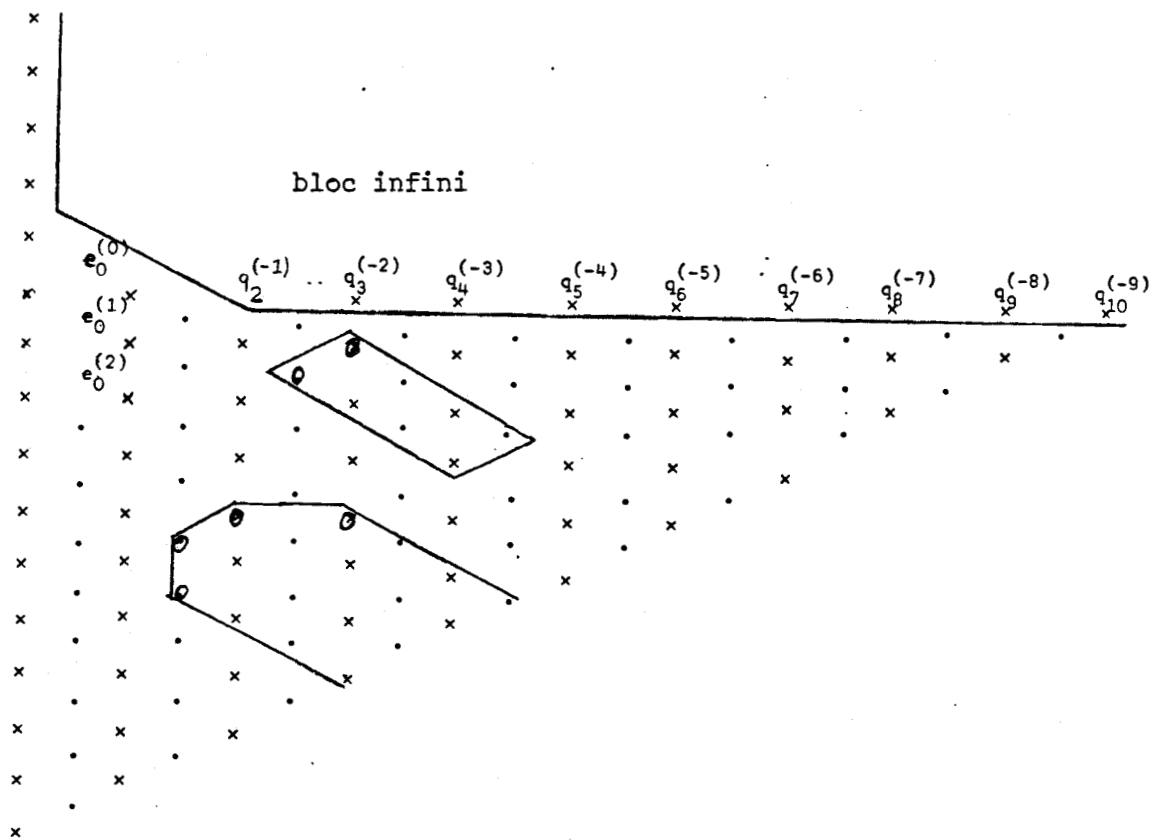
Le tableau



[illegible]

où les numéros de gauche sont les indices des diagonales, les + représentent les polynômes orthogonaux réguliers et les 0 les polynômes dans les blocs, donne l'allure de la table P des polynômes orthogonaux.

De l'allure de la table P on déduit l'allure de la table qd.



Et les éléments de la table qd sont :

Les numéros de gauche sont les indices diagonales.

Les valeurs des éléments appartenant aux blocs qd sont suivies de (*).

INDICE DE COLONNE= 1		INDICE DE COLONNE= 2	
VALEURS DE Q(M,J)	VALEURS DE E(N,J)	VALEURS DE Q(M,J)	VALEURS DE E(N,J)
N=-10	INDEFINI OU INFINI	INDEFINI OU INDEFINI	INDEFINI OU INFINI
N= -9	INDEFINI OU INFINI	INDEFINI OU INFINI	INFINI OU INDEFINI
N= -8	INDEFINI OU INFINI	INDEFINI OU INFINI	INFINI OU INDEFINI
N= -7	INDEFINI OU INFINI	INDEFINI OU INFINI	INFINI OU INDEFINI
N= -6	INDEFINI OU INFINI	INDEFINI OU INFINI	INFINI OU INDEFINI
N= -5	INDEFINI OU INFINI	INDEFINI OU INFINI	INFINI OU INDEFINI
N= -4	INDEFINI OU INFINI	INDEFINI OU INFINI	INFINI OU INDEFINI
N= -3	INDEFINI OU INFINI	INDEFINI OU INFINI	INFINI OU INDEFINI
N= -2	INDEFINI OU INFINI	INDEFINI OU INFINI	INFINI OU INDEFINI
N= -1	INDEFINI OU INFINI	INDEFINI OU INFINI	INFINI OU INDEFINI
N= 0	,9000000000000000+00(*)	-.1000000000000000-01(*)	,1100000000000000+01
N= 1	,8668888888888888+00	-.1111111111111111-01	,1111111111111111+01
N= 2	,8750000000000000+00	-.1388888888888888-01	,1125000000000000+01
N= 3	,8571428571428570+00	-.1785714285714290-01	,8714285714285710+02
N= 4	-.1000000000000000+01	-.1857142857142860+01	-.9100000000000000+02
N= 5	-.1000000000000000+01	,0000000000000000+00(*)	INFINI OU INDEFINI
N= 6	-.1000000000000000+01	INDEFINI OU INFINI	INFINI OU INDEFINI
N= 7	-.8333333333333330+00	,0000000000000000+00(*)	INFINI OU INDEFINI
N= 8	,8000000000000000+00	INDEFINI OU INFINI	INFINI OU INDEFINI
N= 9	,7500000000000000+00	,1666666666666670+00	-.8166666666666660+01
		-.2448979591850730-01	,9775510204081630+01
		-.5000000000000000-01	

INDICE DE COLONNE= 3

INDICE DE COLONNE= 4

VALEURS DE Q(M,J)		VALEURS DE E(N,J)		VALEURS DE Q(M,J)		VALEURS DE E(N,J)	
N=-10	INDEFINI OU INFINI	INFINI OU INDEFINI		INDEFINI OU INFINI		INFINI OU INDEFINI	
N= -9	INDEFINI OU INFINI	INFINI OU INDEFINI		INDEFINI OU INFINI		INFINI OU INDEFINI	
N= -8	INDEFINI OU INFINI	INFINI OU INDEFINI		INDEFINI OU INFINI		INFINI OU INDEFINI	
N= -7	INDEFINI OU INFINI	INFINI OU INDEFINI		INDEFINI OU INFINI		INFINI OU INDEFINI	
N= -6	INDEFINI OU INFINI	INFINI OU INDEFINI		INDEFINI OU INFINI		INFINI OU INDEFINI	
N= -5	INDEFINI OU INFINI	INFINI OU INDEFINI		INDEFINI OU INFINI		INFINI OU INDEFINI	
N= -4	INDEFINI OU INFINI	INFINI OU INDEFINI		INDEFINI OU INFINI		INFINI OU INDEFINI	
N= -3	INDEFINI OU INFINI	INFINI OU INDEFINI		INDEFINI OU INFINI		INFINI OU INDEFINI	
N= -2	-.1100000000000000-01(*)	.1100000000000000+01		-.1210000000000000-01(*)		.9200909090909100+02	
N= -1	.0000000000000000+00(*)	INFINI OU INDEFINI		.9090909090909100+02		-.9509090909090920+02	
N= 0	INDEFINI OU INFINI	INFINI OU INDEFINI		INDEFINI OU INFINI		.1099421467603300+03(*)	
N= 1	-.9218181818181810+02	.1181818181818180+01		-.1089421467603300+03(*)		.1180983710569500+01	
N= 2	.0000000000000000+00(*)	INFINI OU INDEFINI		-.8344712406724370-03		-.7999165528751320+01	
N= 3	INDEFINI OU INFINI	INFINI OU INDEFINI		INDEFINI OU INFINI		.7999901380670600+02(*)	
N= 4	INDEFINI OU INFINI	INFINI OU INDEFINI		INDEFINI OU INFINI			
N= 5	-.7983333333333320+02(*)						



INDICE DE COLONNE= 5

INDICE DE COLONNE= 6

VALEURS DE Q(M,J)		VALEURS DE E(N,J)	
N=-10	INDEFINI OU INFINI	INFINI OU INDEFINI	INDEFINI OU INFINI
N= -9	INDEFINI OU INFINI	INFINI OU INDEFINI	INFINI OU INDEFINI
N= -8	INDEFINI OU INFINI	INFINI OU INDEFINI	INFINI OU INDEFINI
N= -7	INDEFINI OU INFINI	INFINI OU INDEFINI	INFINI OU INDEFINI
N= -6	INDEFINI OU INFINI	INFINI OU INDEFINI	INFINI OU INDEFINI
N= -5	INDEFINI OU INFINI	INFINI OU INDEFINI	INFINI OU INDEFINI
N= -4	-.111531000000000+01(*)	INFINI OU INDEFINI	INFINI OU INDEFINI
N= -3	-.939540649050000+02	INFINI OU INDEFINI	INFINI OU INDEFINI
N= -2	.1156179304603860+01	INFINI OU INDEFINI	INFINI OU INDEFINI
N= -1	.1170241845603400+01	INFINI OU INDEFINI	INFINI OU INDEFINI
N= 0	.5052130158421680-02	INFINI OU INDEFINI	INFINI OU INDEFINI
N= 1	.100009199100000+02	INFINI OU INDEFINI	INFINI OU INDEFINI

INDICE DE COLONNE= 7

INDICE DE COLONNE= 8

VALEURS DE Q(M,J)		VALEURS DE E(N,J)	
N=-10	INDEFINI OU INFINI	INFINI OU INDEFINI	INDEFINI OU INFINI
N= -9	INDEFINI OU INFINI	INFINI OU INDEFINI	INFINI OU INDEFINI
N= -8	INDEFINI OU INFINI	INFINI OU INDEFINI	INFINI OU INDEFINI
N= -7	INDEFINI OU INFINI	INFINI OU INDEFINI	INFINI OU INDEFINI
N= -6	-.2189103100001040+01(*)	INFINI OU INDEFINI	INFINI OU INDEFINI
N= -5	.6016015734299600-01	INFINI OU INDEFINI	INFINI OU INDEFINI
N= -4	-.6124928801610130+01	INFINI OU INDEFINI	INFINI OU INDEFINI
N= -3	-.1074061473703240+01	INFINI OU INDEFINI	INFINI OU INDEFINI

INDICE DE COLONNE= 9

INDICE DE COLONNE=10

	VALEURS DE Q(M,J)	VALEURS DE E(N,J)	VALEURS DE Q(M,J)	VALEURS DE E(N,J)
N=-10	INDEFINI OU INFINI	INFINI OU INDEFINI	INDEFINI OU INFINI	INFINI OU INDEFINI
N= -9	INDEFINI OU INFINI	INFINI OU INDEFINI	INDEFINI OU INFINI	INFINI OU INDEFINI
N= -8	-.1010557171008890+01(*)	.1217748489161490+01	-.1230701888000520+01(*)	
N= -7	.1707425709400700+01			

2° Pour NL = 10, MON = 0, MAX = - 1 (méthode progressive) et les c_i pour $i = 0, \dots, 10$ de l'exemple précédent, les résultats sont :

Le tableau représentant l'allure de la table P (resp. qd) est la partie au Sud du bloc infini de l'exemple précédent d'après la remarque 2.8.

Les numéros de gauche et les valeurs suivies de (*) sont définis comme dans l'exemple précédent.

Q-D PAR METHODE PROGRESSIVE

INDICE DE COLONNE= 2

INDICE DE COLONNE= 1

	VALEURS DE Q(M,J)	VALEURS DE E(N,J)	VALEURS DE Q(M,J)	VALEURS DE E(N,J)
N= 0	.9000J00000000000+00(*)	-.111111111111100-01	.000000000000000+00(*)	.1100000000000010+01
N= 1	.3688688888888890+00	-.1388688888888890-01	.111111111111120+01	.000000000000000+00(*)
N= 2	.8150J00000000000+00	-.1785714285714300-01	.1125J000000000010+01	.8799777777777730+02
N= 3	.8571428571426570+00	-.1857142857142860+01	.8914285714285640+02	-.8099999999999920+02
N= 4	-.1000000000000000+01	.000000000000000+00(*)	.3552713678500000-13(*)	INFINI OU INDEFINI
N= 5	-.1000000000000000+01	.000000000000000+00(*)	INDEFINI OU INFINI	INFINI OU INDEFINI
N= 6	-.1000J00000000000+01	.000000000000000+00(*)	INDEFINI OU INFINI	INFINI OU INDEFINI
N= 7	-.853333333333374530+00	.16666666666685470+00	-.8166666666579320+01	.9775510203989020+01
N= 8	.80000000000005190+00	.1633333333337970+01	-.2448979592826820-01	
N= 9	.7499999999805700+00	-.5000000001994870-01		

INDICE DE COLONNE= 4

INDICE DE COLONNE= 3

	VALEURS DE Q(M,J)	VALEURS DE E(N,J)	VALEURS DE Q(M,J)	VALEURS DE E(N,J)
N= -2	.000000000000000+00(*)	.110000000000000+01	.000000000000000+00(*)	.9200909090909090+02
N= -1	-.9769962616701380-14(*)	INFINI OU INDEFINI	.9090909090909090+02	-.9509090909090910+02
N= 0	INDEFINI OU INFINI	INDEFINI OU INFINI	INDEFINI OU INFINI	-.1099421487603290+03(*)
N= 1	-.92181818181817+02	.11818181818180+01	.1089421487603290+03(*)	.1180983710569490+01
N= 2	.000000000000000+00(*)	INFINI OU INDEFINI	-.3344712486810560-03	-.7999165528662090+01
N= 3	INDEFINI OU INFINI	INFINI OU INDEFINI	INDEFINI OU INFINI	-.7999901380509780+02(*)
N= 4	INDEFINI OU INFINI	INFINI OU INDEFINI	INDEFINI OU INFINI	
N= 5	.7933333333172310+02(*)	INFINI OU INDEFINI		

INDICE DE COLONNE= 5

INDICE DE COLONNE= 6

	VALEURS DE Q(M,J)	VALEURS DE E(N,J)	VALEURS DE Q(M,J)	VALEURS DE E(N,J)
N= -4	.0000000000000000+00(*)	.0000000000000000+00(*)		
N= -3	-.9395406470522190+02	-.1944973996461000+01	.9340376393811840+00	-.1010966357079820+01
N= -2	.1156179384606650+01	.1933519931169020-01	.8403875039733120+00	-.1378493609618180-01
N= -1	.1170241845623390+01	.1406246098453870-01	.1364266117706250+01	.5374410747794780+00
N= 0	.5652130156422470-02	.1639399507452450-01	.6681928222043050+00	-.6596793004914220+00
N= 1	.1000091991076450+02	.1996102251944000+01		

INDICE DE COLONNE= 7

INDICE DE COLONNE= 8

	VALEURS DE Q(M,J)	VALEURS DE E(N,J)	VALEURS DE Q(M,J)	VALEURS DE E(N,J)
N= -6	.0000000000000000+00(*)	.0000000000000000+00(*)		
N= -5	.6616613784402240-01	-.9427982192357960+00	.4531209989978860+00	-.4896772202439110+00
N= -4	-.6124728851533360+01	-.6266881925473570+01	.4674628066033700+01	-.1645174358429760+01
N= -3	-.1674561473767630+01	.4987808452531210+01		

INDICE DE COLONNE= 9

	VALEURS DE Q(M,J)	VALEURS DE E(N,J)
N= -8	.0000000000000000+00(*)	
N= -7	.1707425709429450+01	.1217748489185540+01
N= 0		



Conclusion : Les éléments communs aux tables qd des exemples 1° et 2° et extérieurs aux blocs devraient être égaux, les erreurs de calcul font qu'ils ne le sont qu'à 10^{-10} près.


```

1.      SUBROUTINE QUDE(IR,MOI,MA,BLOC)
2.      C*****
3.      C      CE SOUS-PROGRAMME CALCULE LES COEFFICIENTS Q(M,J) ET E(M,J) DU Q-D
4.      C      PAR LES METHODES NORMALE ET PROGRESSIVE
5.      C      SI Q=MA, METHODE NORMALE ET METHODE PROGRESSIVE
6.      C      SI MA=U, METHODE NORMALE
7.      C      SI MA=V, METHODE PROGRESSIVE
8.      C      DC EST LE TABLEAU DES COEFFICIENTS DE LA SERIE FORMELLE INDICES DE
9.      C      MOI A NL, DC EST TRANSMIS AU SOUS PROGRAMME PAR COMMON
10.     C      MOI ETANT LE NUMERO DE LA PREMIERE DIAGONALE DANS LA TABLE Q-D
11.     C      CE SOUS PROGRAMME UTILISE LES SOUS PROGRAMMES EST ET DETECT
12.     C*****
13.     C***** CARTES POUVANT ETRE CHANGEES PAR L'UTILISATEUR*****
14.     COMMON DC(-20:20,0:20), DC(-20:20), DBR(-20:20,0:20), DQ(-20:20,0:20)
15.     1, DC(-20:20,0:20), NDF(-20:20,0:20)/S/N(20), K(20), NR(20)
16.     DIMENSION NB(-20:20,0:20), DA(-20:20,0:20), DB(-20:20,0:20)
17.     INTEGER BLOC, BLOC1, COL
18.     MAT=10, DEPS=1, D=10
19.     C***** FIN*****
20.     C*****
21.     C      SORTIE DES COEFFICIENTS DE LA SERIE FORMELLE
22.     C*****
23.     DO 117 I=MOI,IR
24.     117 WRITE(MAT,400)I,DC(I)
25.     400 FORMAT(5X,'C(',I3,')=',D22,15)
26.     MAX=MA, MON=MOI, NL=IR, M=NL, IF(MAX)2000,4000,4000
27.     2000 MAX=-1
28.     DO 34 J=0,IR
29.     IF(ABS(DC(J)),GT,1.D-15)GO TO 5002
30.     34 CONTINUE,WRITE(MAT,500);STOP
31.     500 FORMAT(5X,'TOUS LES C(I) SONT NULS SAUF PEUT ETRE LE DERNIER')
32.     C*****
33.     C      CALCUL DES COEFFICIENTS DC DE LA SERIE RECIPROQUE
34.     C*****
35.     5002 NS=J, DO(0,0)=1./DC(NS), NL=IR-NS, MON=-NL, M=NL
36.     IF(NL,LE,0)WRITE(MAT,500);STOP
37.     DO 30 I=1,NL
38.     DC(I,0)=0
39.     DO 36 J=0,I-1
40.     DC(I,0)=DC(I,0)+DC(NS+I-J)*DQ(J,0)
41.     30 DC(I,0)=-DC(I,0)/DC(NS)
42.     C*****
43.     C      SORTIE DES COEFFICIENTS DE LA SERIE RECIPROQUE
44.     C*****
45.     DO 32 I=MON,NL
46.     DC(I)=0; IF(I,GE,0)DC(I)=DQ(I,0);WRITE(MAT,400)I,DC(I)
47.     32 CONTINUE
48.     4000 I=0
49.     2100 DO 2 J=MON,M+2
50.     2 NDF(J,1)=1; IF(M,LE,MON)GO TO 2111
51.     M=M-2; I=I+1; GO TO 2100
52.     2111 DO 1 I=MON,IR
53.     1 NB(I,0)=1; BLOC=0; COL=1; J=MON; WRITE(MAT,252)
54.     C*****
55.     C      RECHERCHE DES BLOCS APPARAISSANT DES LA PREMIERE COLONNE
56.     C*****
57.     IF(ABS(DC(J)),GT,1.D-15)GO TO 75
58.     BLOC=BLOC+1; K(BLOC)=COL; N(BLOC)=J; NR(BLOC)=0; J=J+1
59.     45 IF(ABS(DC(J)),GT,1.D-15)GO TO 75
60.     NR(BLOC)=NR(BLOC)+1; J=J+1; IF(J,EQ,NL)WRITE(MAT,500);STOP
61.     GO TO 45
62.     75 DC(J,1)=DC(J+1)/DC(J); IF(J,GE,NL-1)GO TO 60
63.     IF(ABS(DC(J+1)),GT,1.D-15)J=J+1;GO TO 75
64.     BLOC=B, DC+1; NR(BLOC)=0; N(BLOC)=J+1; K(BLOC)=1; J=J+2
65.     70 IF(J,GE,NL)GO TO 81
66.     IF(ABS(DC(J)),GT,1.D-15)GO TO 75
67.     NR(BLOC)=NR(BLOC)+1; J=J+1;GO TO 70
68.     252 FORMAT(5X,'LES BLOCS SONT:')
69.     752 FORMAT(5X,'NU=',I2,3X,'N=',I4,3X,'K=',I2,3X,'R=',I2)
70.     81 IF(ABS(DC(NL)),LT,1.D-15)NR(BLOC)=NR(BLOC)+1
71.     60 IF(BLOC,LE,0)GO TO 80
72.     C*****
73.     C      DELIMITATION DES BLOCS DEMARRANT A LA PREMIERE COLONNE
74.     C*****
75.     DO 5 L=1,BLOC
76.     WRITE(MAT,752)L,N(L),K(L),NR(L)
77.     DO 5 J=U,NR(L)
78.     DO 5 I=U,NR(L)
79.     5 NDF(N(L)+J-I,K(L)+I)=0
80.     80 DO 8 J=MON,NL-2
81.     IF((NDF(J,1),EQ,0).OR,(NDF(J+1,1),EQ,0))GO TO 8
82.     DC(J,1)=DC(J+1,1)-DQ(J,1)
83.     8 CONTINUE; LIMITE=NL-2
84.     C*****
85.     C      RECHERCHE DES AUTRES BLOCS
86.     C*****
87.     706 I=MON; BLOC1=BLOC
88.     100 IF((NDF(I,COL),EQ,U).OR,(NDF(I+1,COL),EQ,0))GO TO 110

```



```

89. IF(ABS(VE(I,COL)).GT.DEPS)GO TO 110
90. BLOC=BLOC+1;K(BLOC)=COL+1;NR(BLOC)=0;N(BLOC)=I;I=I+1
91. 333 IF(I.GT.LIMITE) GO TO 90
92. IF(NDF(I,COL).EQ.0) GO TO 110
93. IF(ABS(VE(I,COL)).GT.DEPS) GO TO 110
94. DE(I,COL)=0;NR(BLOC)=NR(BLOC)+1;I=I+1;GO TO 333
95. 110 IF(I.GE.LIMITE) GO TO 90
96. I=I+1;GO TO 100
97. 90 IF(BLOC1.GE.BLOC) GO TO 125
98. C*****
99. C DELIMITATION DES AUTRES BLOCS
100. C*****
101. DO 9 J=BLOC1+1,BLOC
102. WRITE(MAT,752)J,N(J),K(J),NR(J)
103. DO 4 L=U,NR(J)
104. DO 9 I=U,NR(J)
105. 9 NDF(N(J)+L-I,K(J)+I)=0
106. 125 IF(LIMITE.LE.MON)GO TO 701
107. COL=COL+1;LIMITE=LIMITE-1;J=MON
108. C*****
109. C CALCUL DES COEFFICIENTS O(N,J)
110. C*****
111. 514 IF((NDF(J+2,COL-2).EQ.0).AND.(NDF(J+1,COL-1).EQ.1))GO TO 510
112. IF(NDF(J,COL).EQ.0) GO TO 511
113. IF(NDF(J+1,COL).EQ.0)JL=J;DQ(J,COL)=0;GO TO 534
114. IF((NDF(J+1,COL-1).EQ.0).OR.(NDF(J+2,COL-1).EQ.0))JL=J;GO TO 508
115. IF(NDF(J,COL-1).EQ.0)JL=J;GO TO 512
116. IF(NDF(J+1,COL-2).EQ.0)GO TO 825
117. DQ(J,COL)=DQ(J+1,COL-1)+DE(J+1,COL-1)/DE(J,COL-1);GO TO 511
118. 512 DQ(J,COL)=DQ(J+1,COL-1)*DE(J+1,COL-1);GO TO 534
119. 508 CALL DETECT(J,1,COL,BLOC,L)
120. CALL EST(N(L),NR(L),K(L),MON,LIMITE);J=N(L)-1;GO TO 534
121. 510 CALL DETECT(J,2,COL,BLOC,L);IF(J.GT.N(L)-NR(L)-2) GO TO 830
122. IF(NDF(J,COL).EQ.0)J=J+1;GO TO 830
123. IF(NDF(J+1,COL).EQ.0)DQ(J,COL)=0;J=J+1;GO TO 830
124. DQ(J,COL)=-DCR(J+1,COL)/DE(J,COL-1);J=J+1
125. 830 IF(J.GT.LIMITE) GO TO 513
126. IF(J.GE.N(L)-1)J=J-1;GO TO 511
127. IF(NDF(J,COL).EQ.0)J=J+1;GO TO 830
128. IF(NDF(J+1,COL).EQ.0)DQ(J,COL)=0;J=J+1;GO TO 830
129. DQ(J,COL)=DCR(J+1,COL)/DCR(J,COL);J=J+1;GO TO 830
130. 825 DQ(J,COL)=-DQ(J+1,COL-1)+DE(J+1,COL-1)/DCR(J,COL);GO TO 511
131. 534 IF(JL.LE.MON) GO TO 511
132. IF((NDF(JL-1,COL).EQ.0).OR.(NDF(JL,COL-1).GT.0))GO TO 511
133. JL=JL-1;CALL DETECT(JL,1,COL,BLOC,L);NR1=NR(L);N1=N(L);K1=K(L)
134. IF((K1.LT.2).OR.(NDF(N1+1,K1-2).EQ.0))GO TO 522
135. DCR(N1,K1+NR1+1)=DCR(N1+1,K1+NR1)-DQ(N1,K1+NR1+1);GO TO 532
136. 522 DCR(N1,K1+NR1+1)=-DQ(N1,K1+NR1+1)
137. 532 IF(NDF(N1-1,K1-1).EQ.0)GO TO 528
138. DCR(N1-1,K1+NR1+2)=DCR(N1,K1+NR1+1)-DCR(N1-1,K1+NR1+1);GO TO 527
139. 528 DCR(N1-1,K1+NR1+2)=DCR(N1,K1+NR1+1)
140. 527 DE(N1-1,K1+NR1+1)=DCR(N1-1,K1+NR1+2)
141. 511 IF(J.GE.LIMITE) GO TO 513
142. J=J+1;GO TO 514
143. 513 LIMITE=LIMITE-1;J=MON;IF(LIMITE.LT.MON)GO TO 701
144. C*****
145. C CALCUL DES COEFFICIENTS E(N,J)
146. C*****
147. 608 IF((NDF(J+2,COL-2).EQ.0).AND.(NDF(J+1,COL-1).EQ.1))GO TO 604
148. IF((NDF(J,COL).EQ.0).OR.(NDF(J+1,COL).EQ.0))GO TO 600
149. IF((NDF(J+2,COL-1).EQ.0).OR.(NDF(J+1,COL-1).EQ.0))GO TO 602
150. IF(NDF(J,COL-1).EQ.0) GO TO 603
151. DE(J,COL)=DQ(J+1,COL)-DQ(J,COL)+DE(J+1,COL-1);GO TO 600
152. 603 DE(J,COL)=DQ(J+1,COL)+DE(J+1,COL-1);GO TO 600
153. 602 CALL DETECT(J,1,COL,BLOC,L);J=N(L)-1;GO TO 600
154. 604 CALL DETECT(J,2,COL,BLOC,L)
155. 1111 IF((J.GT.LIMITE).OR.(J.GT.N(L)-2))J=J-1;GO TO 600
156. IF((NDF(J,COL).EQ.0).OR.(NDF(J+1,COL).EQ.0))J=J+1;GO TO 1111
157. DE(J,COL)=DQ(J+1,COL)-DQ(J,COL);J=J+1;GO TO 1111
158. 600 IF(J.GE.LIMITE) GO TO 706
159. J=J+1;GO TO 608
160. 701 IF(MAX.LT.0)NL=IR;LIMITE=NL-NS-1;M=1;J=1;GO TO 2004
161. C*****
162. C SORTIE DE LA TABLE O-D CALCULEE PAR LA METHODE NORMALE
163. C*****
164. MAX=MAX-1;LIMITE=NL-1;I=1;J=MON;WRITE(MAT,2001);WRITE(MAT,1003)
165. 1003 FORMAT(//)
166. 1002 FORMAT(//)
167. 625 FORMAT(10X,' VALEURS DE O(M,J)',14X,' VALEURS DE E(N,J)')
168. 866 WRITE(MAT,1003);WRITE(MAT,1000);WRITE(MAT,1002)
169. WRITE(MAT,625);WRITE(MAT,1002)
170. 806 IF(NDF(J,I).EQ.0) GO TO 702
171. IF(NDF(J+1,I-1).EQ.0) GO TO 704
172. WRITE(MAT,807)J,DQ(J,I);IF(J.GE.LIMITE)GO TO 900
173. IF(NDF(J+1,I).EQ.0) GO TO 815
174. WRITE(MAT,807)DE(J,I);GO TO 814
175. 704 WRITE(MAT,850)J;IF(J.GE.LIMITE)GO TO 900
176. WRITE(MAT,807)DE(J,I);GO TO 814

```

```

177. 702 WRITE(MAT,850)J,IF(J.GE.LIMITE)GO TO 900
178. 815 WRITE(MAT,850)
179. 814 IF(J.GE.LIMITE) GO TO 900
180. J=J+1;GO TO 806
181. 900 LIMITE=LIMITE-2;IF(LIMITE.LT.MON)I=0;LIMITE=NL+1;GO TO 936
182. I=I+1;I=MON;GO TO 866
183. C*****
184. C SCHEMATISATION DE LA TABLE -P- QUAND LA TABLE Q-D EST CALCULEE PAR
185. LA METHODE NORMALE
186. C*****
187. 936 DO 48 I=MON,LIMITE
188. IF(NDF(J,I).EQ.1)NB(J,I)=1H+1GO TO 48
189. NB(J,I)=1H0
190. 48 CONTINUE;LIMITE=LIMITE-2;IF(LIMITE.LT.MON)ME=NL/2;GO TO 935
191. I=I+1;GO TO 936
192. 935 WRITE(MAT,800)MON;IF(MON.LT.0)ME=0
193. DO 38 I=MON,ME;WRITE(MAT,1002);KT=J+1
194. 38 WRITE(MAT,800)(KT,(NB(J-I,I),I=0,J-MON))
195. DO 39 J=1,NL-ME;IF(ME+J.GT.NL)GO TO 862
196. WRITE(MAT,1002);KT=ME+J+1
197. 39 WRITE(MAT,800)(KT,(NB(ME+J-I,I),I=0,NL-ME-J+1))
198. 862 IF(MAX.GE.0) GO TO 2000
199. RETURN
200. C*****
201. C SYMETRISATION EN VUE D'ORTENIR LA TABLE Q-D PAR LA METHODE PROGRESSIVE
202. C*****
203. 2004 IF(NDF(1-J,M+J-1).EQ.0) GO TO 2011
204. IF(NDF(2-J,J+M-1).EQ.0) GO TO 2010
205. NB(NS+J,M)=1;DA(NS+J,M)=DE(1-J,M+J-1)
206. IF(NDF(1-J,M+J).EQ.0) GO TO 2013
207. DB(NS+1,M)=DC(1-J,J+M);GO TO 2013
208. 2011 IF(NDF(1-J,J+M).EQ.0).OR.(NDF(2-J,J+M-1).EQ.0)GO TO 2010
209. NB(NS+1,M)=1;DB(NS+J,M)=DC(1-J,M+J);GO TO 2013
210. 2010 NB(NS+1,M)=0
211. 2013 IF(J.GE.LIMITE) GO TO 2014
212. J=J+1;GO TO 2004
213. 2014 IF((NDF(1-J,M+J).EQ.0).OR.(NDF(1-J,M+J).EQ.0))GO TO 2017
214. NB(NS+J+1,M)=1;DA(NS+J+1,M)=DE(-J,M+J);GO TO 2018
215. 2017 NB(NS+J+1,M)=0
216. 2018 IF(LIMITE.LE.1)LIMITE=NL-NS-1;M=1;J=1;GO TO 3500
217. LIMITE=LIMITE-1;M=M+1;J=1;GO TO 2004
218. 3500 NB(NS+1-J,M+J-1)=1;IF(NDF(J,M).EQ.0)GO TO 3501
219. DB(NS+1-J,M+J-1)=DC(J,M)
220. 3501 IF(J.GE.LIMITE)LIMITE=LIMITE-1;M=1;J=1;GO TO 3004
221. J=J+1;GO TO 3500
222. 3004 IF(NDF(J,M).EQ.0) GO TO 3001
223. IF(NDF(J+1,M).EQ.0) GO TO 3000
224. NB(NS+1-J,M+J)=1;DA(NS+1-J,M+J)=DE(J,M)
225. IF(NDF(J,M+1).EQ.0) GO TO 3002
226. DB(NS+1-J,M+J)=DC(J,M+1);GO TO 3002
227. 3001 IF(NDF(J,M+1).EQ.0).OR.(NDF(J+1,M).EQ.0)GO TO 3000
228. NB(NS+1-J,M+J)=1;DB(NS+1-J,M+J)=DC(J,M+1);GO TO 3002
229. 3000 NB(NS+1-J,M+J)=0
230. 3002 IF(J.GE.LIMITE) GO TO 3003
231. J=J+1;GO TO 3004
232. 3003 IF(LIMITE.LE.1)DA(NS,1)=-DC(1)/DC(0);LIMITE=NL-1;GO TO 3005
233. LIMITE=LIMITE-1;M=M+1;J=1;GO TO 3004
234. 3005 DO 56 I=1,NL-NS-1
235. NB(NS-I,I+1)=1
236. 56 DA(NS-I,I+1)=0;WRITE(MAT,1002);WRITE(MAT,1003);I=1;KA=NS
237. WRITE(MAT,5001);NB(NS,0)=1H+
238. C*****
239. C SORTIE DE LA TABLE Q-D CALCULEE PAR LA METHODE PROGRESSIVE
240. C*****
241. 3012 WRITE(MAT,1000)I;J=KA;WRITE(MAT,1002)
242. WRITE(MAT,820);WRITE(MAT,1002)
243. 3011 IF(NB(J,I).EQ.0) GO TO 3007
244. IF(NB(J+1,I-1).EQ.0) GO TO 3008
245. WRITE(MAT,801)J,DA(J,I);IF(J.GE.LIMITE)GO TO 3010
246. IF(NB(J+1,I).EQ.0) GO TO 3009
247. WRITE(MAT,804)DB(J,I);GO TO 3010
248. 3007 WRITE(MAT,850)J,IF(J.GE.LIMITE) GO TO 3010
249. 3009 WRITE(MAT,850);GO TO 3010
250. 3008 WRITE(MAT,850)J,IF(J.GE.LIMITE)GO TO 3010
251. WRITE(MAT,804)DB(J,I)
252. 3010 IF(J.GE.LIMITE)LIMITE=LIMITE-2;GO TO 3030
253. J=J+1;GO TO 3011
254. 3030 IF(LIMITE.LT.KA)LIMITE=NL-NS;GO TO 3026
255. KA=KA-1;I=I+1;GO TO 3012
256. C*****
257. C SCHEMATISATION DE LA TABLE -P- QUAND LA TABLE Q-D EST CALCULEE PAR
258. LA METHODE PROGRESSIVE
259. C*****
260. 3026 DO 49 I=NS+1,NL+1
261. DO 47 I=0,LIMITE;IF(NB(I-J,J).EQ.1)NB(I-J,J)=1H+1GO TO 47
262. NB(I-J,J)=1H0
263. 47 CONTINUE
264. 49 LIMITE=LIMITE-1

```

BUS
LILLE


```

265.      WRITE(MAT,800)NS;WRITE(MAT,1002);KA=NS+1;WRITE(MAT,800)KA,NB(NS,0)
266.      DO 61 I=1,NL-NS;WRITE(MAT,1002);KA=NS+I+1
267.      61 WRITE(MAT,800)(KA,(NB(NS+I-J,J),J=0,NL-NS-I+1))
268.      801 FORMAT(3X,'N=',I3,3X,D22.15)
269.      856 FORMAT(36X,'INFINI OU INDEFINI',/)
270.      809 FORMAT(34X,D22.15,/)
271.      850 FORMAT(3X,'N=',I3,5X,'INDEFINI OU INFINI')
272.      5001 FORMAT(9X,' W-D PAR METHODE PROGRESSIVE ')
273.      1000 FORMAT(//,20X,'INDICE DE COLONNE=',I2,/)
274.      800 FORMAT(1X,'N=',I3,5X,20(A1,5X))
275.      2001 FORMAT(4X,' W-D PAR METHODE NORMALE')
276.      RETURN
277.      END

```

```

1. SUBROUTINE EST(N,NR,K,MON,MAX)
2. C*****
3. C CE SOUS PROGRAMME CALCULE LES TERMES C(M,J),B(M,J),ET SUIVANT LES CAS
4. C E(N-NR-2,K+NR+1) ET C(N-NR-2,K+NR+1)
5. C LE TRIPLET(N,K,NR) CORRESPOND AUX COORDONNEES DU BLOC CONSIDERE
6. C MON=NUMERO DE LA PREMIERE DIAGONALE DANS LA TABLE DU 0-0
7. C MAX=NUMERO DE LA DIAGONALE OU ON ARRETE LES CALCULS POUR LA COLONNE
8. C QUE L'ON EXPLOITE
9. C*****
10. C*****CARTES POUVANT ETRE CHANGEES PAR L'UTILISATEUR*****
11. COMMONDCR(-20:20,0:20),DC(-20:20),DBR(-20:20,0:20),DQ(-20:20,0:20)
12. 1,DE(-20:20,0:20),NDF(-20:20,0:20)
13. C*****FIN*****
14. C*****
15. C CALCUL DES COEFFICIENTS -C- ET -B- QUI INTERVIENNENT DANS LA RELATION
16. C DE RECURRENCE A TROIS TERMES
17. C*****
18. IF(K.GT.1) GO TO 120
19. DCR(N,K+NR+1)=0;DCR(N-1,K+NR+1)=-DC(N+NR+1)/DC(N-1);J=0
20. 1 IF((N-NR-1+J.GT.MAX).OR.(J.GT.NR))J=NR-1;GO TO 6
21. DBR(N-NR-1+J,K+NR+1)=-DQ(N+NR+1-J,K+J);J=J+1;GO TO 1
22. 120 IF(NDF(N+NR+2,K-2).EQ.0) GO TO 2
23. DCR(N+NR+1,K)=-DE(N+NR+1,K-1)+DQ(N+NR+1,K-1)
24. IF(N-NR-1.GT.MAX)J=1;GO TO 3
25. DBR(N-NR-1,K+NR+1)=-DQ(N+NR+1,K)-DE(N+NR+1,K-1);J=1;GO TO 3
26. 2 IF(N-NR-1.GT.MAX)J=1;GO TO 3
27. DBR(N-NR-1,K+NR+1)=-DQ(N+NR+1,K);J=1
28. 3 IF(NDF(N+NR+2-J,K-2).EQ.0) GO TO 4
29. DCR(N+NR+1-J,K+J)=DCR(N+NR+2-J,K+J-1)+DQ(N+NR+1-J,K-1)
30. IF((N-NR-1+J.GT.MAX).OR.(N-NR+J.GT.N))GO TO 5
31. DBR(N-NR-1+J,K+NR+1)=DCR(N+NR+2-J,K+J-1)-DQ(N+NR+1-J,K+J);GO TO 5
32. 4 DCR(N+NR+1-J,K+J)=DCR(N+NR+2-J,K+J-1)
33. IF((N-NR-1+J.GT.MAX).OR.(N-NR+J.GT.N))GO TO 5
34. DBR(N-NR-1+J,K+NR+1)=-DQ(N+NR+1-J,K+J)
35. 5 IF(J.GE.NR+1)J=NR;GO TO 6
36. J=J+1;GO TO 3
37. 6 IF((J.LT.0).OR.(N-NR-1+J.LT.MON)) GO TO 8
38. IF(NDF(N-NR-1+J,K+NR-1-J).EQ.0)GO TO 7
39. IF(NDF(N-NR+J,K+NR-2-J).EQ.0)DA=DCR(N-NR-1+J,K+NR-J);GO TO 7
40. DA=-DE(N-NR-1+J,K+NR-1-J)
41. 7 DCR(N-NR-1+J,K+NR+1)=-DCR(N-NR+J,K+NR+1)/DA;J=J-1;GO TO 6
42. 7 DCR(N-NR-1+J,K+NR+1)=DCR(N-NR+J,K+NR+1);J=J-1;GO TO 6
43. 8 IF(N-NR-2.LT.MON)J=0;GO TO 13
44. C*****
45. C CALCUL DE Q OU E
46. C*****
47. IF(NDF(N-NR-2,K+NR).EQ.0)GO TO 11
48. IF(NDF(N-NR-1,K+NR-1).EQ.0)DA=DCR(N-NR-2,K+NR+1);GO TO 12
49. DA=-DE(N-NR-2,K+NR)
50. 12 DQ(N-NR-2,K+NR+1)=DCR(N-NR-1,K+NR+1)/DA
51. IF(N-NR-1.GT.MAX)RETURN
52. DE(N-NR-2,K+NR+1)=-DBR(N-NR-1,K+NR+1)-DQ(N-NR-2,K+NR+1)
53. J=0;GO TO 13
54. 11 DQ(N-NR-2,K+NR+1)=-DCR(N-NR-1,K+NR+1)
55. IF(N-NR-1.GT.MAX)RETURN
56. DE(N-NR-2,K+NR+1)=-DBR(N-NR-1,K+NR+1);J=0
57. 13 IF(N-NR-1.LT.MON)J=MON-N+NR+1
58. C*****
59. C CALCUL DES AUTRES COEFFICIENTS INTERVENANT DANS LA RELATION DE RECURRENCE
60. C A TROIS TERMES
61. C*****
62. 15 IF((N-NR+J.GT.MAX).OR.(J.GE.NR))RETURN
63. IF(NDF(N-NR-1+J,K+NR-1-J).NE.0)GO TO 20
64. DCR(N-NR-1+J,K+NR+2)=DBR(N-NR+J,K+NR+1);GO TO 14
65. 20 DCR(N-NR-1+J,K+NR+2)=DBR(N-NR+J,K+NR+1)-DCR(N-NR+J-1,K+NR+1)
66. 14 DE(N-NR-1+J,K+NR+1)=DCR(N-NR-1+J,K+NR+2);J=J+1;GO TO 15
67. END

```

```

1. SUBROUTINE DETECT(I,MET,COL,BLOC,L)
2. C*****
3. C CE SOUS PROGRAMME DETERMINE LE NUMERO -L- DU BLOC A L'EST DUQUEL
4. C LA DIAGONALE -I- ET LA COLONNE -COL- SE COUPENT
5. C BLOC=NUMERO DE BLOC DEJA DETERMINEES DANS LA TABLE DE 0-0
6. C MET=1,SI ON EXPLORE LA 1-ERE COLONNE
7. C =2,SI ON EXPLORE LA 2-IEME COLONNE
8. C*****
9. C*****CARTES POUVANT ETRE CHANGEES PAR L'UTILISATEUR*****
10. COMMON/S/N(20),K(20),NR(20)
11. C*****FIN*****
12. INTEGERBLOC,COL
13. DO 21 J=1,BLOC
14. IF((K(J)+NR(J)+MET.LT.COL).OR.(K(J)+NR(J)+MET.GT.COL)) GO TO 21
15. IF((I.LT.N(J)-NR(J)-2).OR.(I.GT.N(J)-MET)) GO TO 21
16. L=J;RETURN
17. 21 CONTINUE
18. END

```

TROISIEME PARTIE

SYSTEMES ADJACENTS DE POLYNOMES ORTHOGONAUX
ET APPROXIMANTS DE PADE
I. SYSTEMES ADJACENTS DE POLYNOMES ORTHOGONAUX ET LEURS ASSOCIES

Dans cette partie on suppose connus les éléments de la table q_d et les polynômes unitaires $\omega_i^{(n)}(x)$ qui interviennent dans les relations de récurrence.

Soit $(c_i)_{i \in \mathbb{N}}$ une suite de nombres réels. On peut définir une fonctionnelle linéaire c telle que $c(x^j) = c_j \quad \forall j \in \mathbb{N}$.

On rappelle que l'ensemble $\{P_k^{(n)}(x)\}_{k \in \mathbb{N}}$ des polynômes orthogonaux par rapport aux fonctionnelles linéaires $c^{(n)}$ définies par

$$\forall n \quad c^{(n)}(x^j) = c_{n+j} \quad \forall j \in \mathbb{N}$$

est appelé systèmes adjacents des polynômes orthogonaux. Ces polynômes sont rangés dans un tableau que l'on appelle table P.

On sait que les entiers h_ℓ et $p_{\ell+1}$ sont définis tels que $H_{h_\ell+1}^{(0)} \neq 0$, $H_{p_{\ell+1}+1}^{(0)} \neq 0$ et $H_i^{(0)} = 0$ pour $i \in \mathbb{N}$, $h_\ell+2 \leq i \leq p_{\ell+1}$ où $\ell+1$ est le numéro de ce groupe de déterminant(s) $H_i^{(0)}$ nul(s) par rapport à la diagonale zéro.

La propriété suivante donne les conditions d'existence des polynômes orthogonaux par rapport à une fonctionnelle linéaire c .

Propriété 3.1. [1, page 41] : Soit ℓ tel que

$$\begin{cases} H_i^{(0)} \neq 0 \text{ pour } i \in \mathbb{N}, p_{\ell}+1 \leq i \leq h_{\ell}+1 \\ H_i^{(0)} = 0 \text{ pour } i \in \mathbb{N}, h_{\ell}+2 \leq i \leq p_{\ell+1} \\ H_{p_{\ell+1}+1}^{(0)} \neq 0 \end{cases}$$

Alors

i) Pour $i \in \mathbb{N}, p_{\ell}+1 \leq i \leq h_{\ell}+1$, $P_i^{(0)}(x)$ existe. Il est unique si l'on fixe le coefficient de x^i .

ii) $P_i^{(0)}(x)$ existe pour $i \in \mathbb{N}, h_{\ell}+2 \leq i \leq h_{\ell} + \text{entier}(\frac{p_{\ell+1}+1-h_{\ell}}{2})$ avec

$$P_i^{(0)}(x) = P_{h_{\ell}+1}^{(0)}(x) \omega_{i-h_{\ell}-1}^{(0)}(x) \text{ où } \omega_{i-h_{\ell}-1}^{(0)}(x) \text{ est un polynôme}$$

unitaire de degré $i-h_{\ell}-1$ exactement.

iii) $P_i^{(0)}(x)$ n'existe pas pour $i \in \mathbb{N}, h_{\ell}+1 + \text{entier}(\frac{p_{\ell+1}+1-h_{\ell}}{2}) \leq i \leq p_{\ell+1}$

iv) $P_{p_{\ell+1}+1}^{(0)}(x)$ existe. Il est unique si l'on fixe le coefficient de $x^{p_{\ell+1}+1}$.

Les polynômes $Q_k^{(n)}(t)$ définis par

$$Q_k^{(n)}(t) = c_k^{(n)} \left(\frac{P_k^{(n)}(x) - P_k^{(n)}(t)}{x - t} \right), \quad c_k^{(n)} \text{ agissant sur la variable } x,$$

sont appelés polynômes associés aux polynômes orthogonaux $P_k^{(n)}(x)$. On sait que les polynômes associés $Q_k^{(n)}(t)$ satisfont la même relation de récurrence à trois termes que les polynômes orthogonaux $P_k^{(n)}(x)$.

Si $P_{i+1}^{(n)}(x)$ existe et est orthogonal régulier, d'après les propriétés 2.4 et 2.6 de [1, pages 102 et 121] on a :

1° Si la diagonale n n'est pas la première diagonale supérieure de la table P

$$P_{i+1}^{(n)}(x) = \omega_{i+1-\text{pr}(i+2,n-1)}^{(n-1)}(x) P_{\text{pr}(i+2,n-1)}^{(n-1)}(x) + E_{i+2}^{(n-1)} P_{\text{pr}(\text{pr}(i+2,n-1),n)}^{(n)}(x)$$

et

(1)

$$Q_{i+1}^{(n)}(t) = \omega_{i+1-\text{pr}(i+2,n-1)}^{(n-1)}(t) (t Q_{\text{pr}(i+2,n-1)}^{(n-1)}(t) - c_{n-1} P_{\text{pr}(i+2,n-1)}^{(n-1)}(t)) + E_{i+2}^{(n-1)} Q_{\text{pr}(\text{pr}(i+2,n-1),n)}^{(n)}(t)$$

$$\text{avec } E_{i+2}^{(n-1)} = \begin{cases} -e_{i+1}^{(n-1)} & \text{si } P_i^{(n)}(x) \text{ est orthogonal régulier} \\ c_{i+2}^{(n-1)} & \text{sinon} \end{cases}$$

2° Si $P_i^{(n+1)}(x)$ est orthogonal régulier

$$P_{i+1}^{(n)}(x) = x P_i^{(n+1)}(x) - q_{i+1}^{(n)} P_{\text{pr}(i+1,n)}^{(n)}(x)$$

et

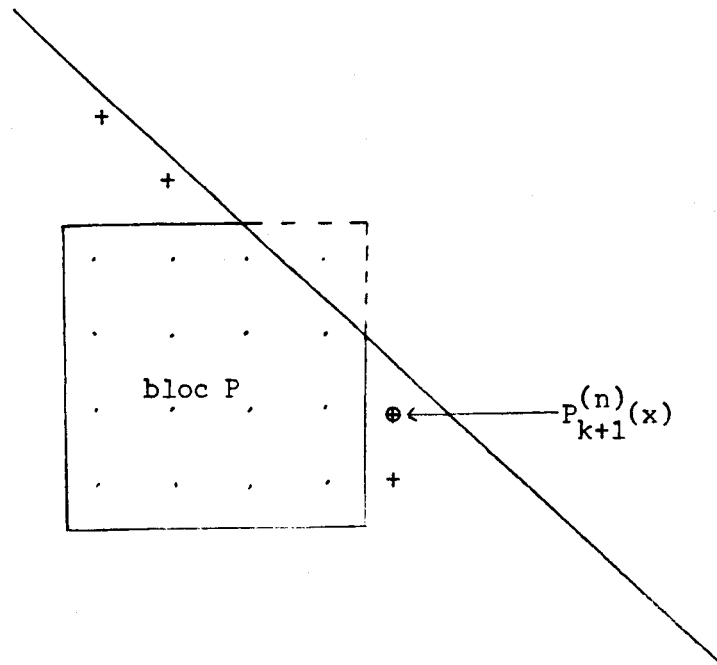
(2)

$$Q_{i+1}^{(n)}(t) = Q_i^{(n+1)}(t) - q_{i+1}^{(n)} Q_{\text{pr}(i+1,n)}^{(n)}(t) + c_n P_i^{(n+1)}(t).$$

Remarque 3.2. : On utilisera la relation de récurrence à trois termes

$$U_{k+1}^{(n)}(x) = (x \omega_{k-\text{pr}(k+1,n)}^{(n)}(x) + B_{k+1}^{(n)}) U_{\text{pr}(k+1,n)}^{(n)}(x) + C_{k+1}^{(n)} U_{\text{pr}(\text{pr}(k+1,n),n)}^{(n)}(x) \quad (3)$$

si les conditions d'application de (1) et (2) ne sont pas satisfaites. C'est-à-dire si le polynôme $P_{k+1}^{(n)}(x)$ existe et est sur la première diagonale supérieure de la table P et à l'Est d'un bloc comme l'indique la figure suivante :



Pour le démarrage des relations (1), (2) et (3) on pose $\forall n \in \mathbb{N}$,

$P_{-1}^{(n)}(x)$ (resp. $P_0^{(n)}(x)$) orthogonal régulier et $P_{-1}^{(n)}(x) = 0$,

$P_0^{(n)}(x) = 1$, $Q_{-1}^{(n)}(t) = 1$ et $Q_0^{(n)}(t) = 0$.

Définition 3.3. : On dira que le polynôme $P_k^{(m)}(x)$ est sur la ligne ℓ ou appartient à la ligne numéro ℓ si

$$\ell = k+m.$$

On peut diviser la table P des polynômes orthogonaux en deux demi-tables : les demi-tables supérieure et inférieure, la demi-table supérieure étant constituée des lignes qui ont un polynôme sur la première diagonale supérieure de la table P.

ALGORITHME DE CALCUL DES POLYNOMES ORTHOGONAUX

REGULIERS ET LEURS ASSOCIES

On utilise les relations (1), éventuellement les relations (3), si $P_k^{(n)}$ est à l'Est d'un bloc, et les relations (2) dans les autres cas.

On notera $[x]$ la partie entière de x .

Dans cet algorithme l correspond au numéro de la ligne et j au degré du polynôme en P considéré. On suppose que l'indice de la première diagonale supérieure de la table P est zéro.

Le passage d'un numéro au suivant se fait toujours en séquence, sauf s'il y a un ordre aller en.

Soit à calculer la table P pour les c_i $i = 0, \dots, n$.

A - Calcul de la demi-table supérieure

0) On initialise la ligne zéro

$$P_{-1}^{(1)}(x) = 0, Q_{-1}^{(1)}(t) = 1$$

$$P_0^{(0)}(x) = 1, Q_0^{(0)}(t) = 0$$

et on pose $l = 1$ et $M = [n+1/2]$

1) Si $l > M$ alors $l-1 \rightarrow l$ et aller en B

Initialisation de la ligne l

$$P_{-1}^{(l+1)}(x) = 0, Q_{-1}^{(l+1)}(t) = 1$$

$$P_0^{(l)}(x) = 1, Q_0^{(l)}(t) = 0$$

$1 \rightarrow j$

2) Si le polynôme $P_j^{(\ell-j)}(x)$ n'est pas orthogonal régulier, aller en 5.

Si le polynôme $P_{j-1}^{(\ell-j+1)}$ n'est pas orthogonal régulier, alors $P_j^{(\ell-j)}(x)$

est à l'Est d'un bloc, aller en 3.

Sinon

$$P_j^{(\ell-j)}(x) = x P_{j-1}^{(\ell-j+1)}(x) - q_j^{(\ell-j)} P_{pr(j, \ell-j)}^{(\ell-j)}(x)$$

et

$$Q_j^{(\ell-j)}(t) = Q_{j-1}^{(\ell-j+1)}(t) - q_j^{(\ell-j)} Q_{pr(j, \ell-j)}^{(\ell-j)}(t) + c_{\ell-j} P_{j-1}^{(\ell-j+1)}(t),$$

aller en 5.

3) Cas où le polynôme $P_j^{(\ell-j)}(x)$ est à l'Est d'un bloc

3.1. Si $\ell-j = 0$ alors $P_j^{(\ell-j)}$ est sur la première diagonale supérieure et à l'Est d'un bloc, aller en 4

Sinon on a :

$$P_j^{(\ell-j)}(x) = P_j^{(\ell-j-1)}(x) + E_{j+1}^{(\ell-j-1)} P_{pr(j, \ell-j)}^{(\ell-j)}(x)$$

et

$$Q_j^{(\ell-j)}(t) = t Q_j^{(\ell-j-1)}(x) - c_{\ell-j-1} P_j^{(\ell-j-1)}(t) + E_{j+1}^{(\ell-j-1)} Q_{pr(j, \ell-j)}^{(\ell-j)}(t)$$

aller en 5.

4) On utilise la relation de récurrence à trois termes pour calculer $P_j^{(\ell-j)}(x)$ et $Q_j^{(\ell-j)}(t)$.

5) Si $j < \ell$, $j+1 \rightarrow j$ et aller en 2

$\ell+1 \rightarrow \ell$ et aller en 1.

B - Calcul de la demi-table inférieure

On utilise le même algorithme que pour la demi-table supérieure sans 3.1. et 4), et avec

0) Si n est impair $M = M-1$.

M est le degré du dernier polynôme de la ligne numéro ℓ .

1) $\ell+1 \rightarrow \ell$

si $\ell > n$ fin

Initialisation de la ligne ℓ

$$P_{-1}^{(\ell+1)}(x) = 0, Q_{-1}^{(\ell+1)}(t) = 1, P_0^{(\ell)}(x) = 1 \text{ et } Q_0^{(\ell)}(t) = 0$$

$1 \rightarrow j$

et

5) Si $j < M$, $j+1 \rightarrow j$ et aller en 2

$M = M-1$ et aller en 1.

Remarque : On peut indiquer les diagonales de P à partir de n'importe quel entier, évidemment le calcul de la table P n'est possible que si les indices des diagonales de P ont un sens pour la table q_d .

II. APPROXIMANTS DE PADE

Les approximations de Padé ont été définies dans la partie I. Nous allons donner un théorème, qui généralise le théorème 1.21, et la propriété des approximations de Padé dans un bloc.

Théorème 3.4. : On suppose $n+k-1 \geq 0$.

i) Si le polynôme $P_k^{(n)}$ est orthogonal régulier, alors l'approximant de Padé $[n+k-1/k]_f(t)$ existe et, est unique et égal à

$$\sum_{j=0}^{n-1} c_j t^j + t^n \frac{Q_k^{(n)}(t)}{P_k^{(n)}(t)}$$

- ii) Si $P_k^{(n)}$ est orthogonal singulier, alors l'approximant de Padé $[n+k-1/k]_f(t)$ existe, est unique et est égal à

$$\sum_{j=0}^{n-1} c_j t^j + t^n \frac{Q_{pr(k,n)}^{(n)}(t)}{P_{pr(k,n)}^{(n)}(t)}, \text{ autrement dit, il est identique à } [n+pr(k,n)-1/pr(k,n)]_f(t).$$

- iii) Si $P_k^{(n)}$ est quasi-orthogonal, l'approximant de Padé n'existe pas.

Rappel : $P_k^{(n)}(t) = t^k P_k^{(n)}(t^{-1})$ et $Q_k^{(n)}(t) = t^{k-1} Q_k^{(n)}(t^{-1})$

Propriété 3.5. : Dans un bloc Padé, tous les approximants de Padé situés au dessus de l'antidiagonale qui correspond à l'antidiagonale principale du bloc P sont identiques quand on les a mis sous la forme irréductible.

III. MISE EN OEUVRE

La mise en oeuvre s'effectue avec les sous-programmes de calcul de la table qd et des sous-programmes APADE et ECRIRE. Dans le sous-programme APADE on note

$P_k^{(j)}(x)$ par $P(J, K)$ et les $P(J, I)$, pour $0 \leq I \leq K$, sont les coefficients du polynôme $P_k^{(j)}(x)$, $P(J, I)$ étant le coefficient de x^I .

$P_{pr(k,j)}^{(j)}(x)$ par $PR(J, K)$

$Q_k^{(j)}(t)$ par $Q(J, K)$

$Q_{pr(k,j)}^{(j)}(t)$ par $QR(J, K)$

et leurs coefficients sont notés de la même manière que ceux de $P_k^{(j)}(x)$.

Les coefficients de la série formelle, les éléments de la table qd, les $B_k^{(j)}$ et les $C_k^{(j)}$ sont échangés entre programme principal et sous-programmes par COMMON.

MODE D'EMPLOI DES SOUS-PROGRAMMES

* * * APADE * * * *

Appel : CALL APADE(NL)

Paramètres : Les coefficients de la série formelle sont transmis à ce sous-programme par COMMON, ils sont rangés dans un tableau de longueur NL+1, indicé de 0 à NL et donc l'indice de la première diagonale supérieure de la table P est zéro.

NL paramètre d'entrée.

Sous-programmes utilisés : QUDE, DETECT, EST et, ECRIRE si l'on désire calculer les approximants de Padé.

Description : Ce sous-programme calcule les polynômes orthogonaux réguliers et leurs associés. Il fait éventuellement appel à ECRIRE pour le calcul de l'approximant de Padé dès qu'un polynôme et son associé sont connus.

* * * * ECRIRE * * * * *

Appel : CALL ECRIRE(R, T, IA, J)

Paramètres : T est le tableau des coefficients du polynôme $P_J^{(IA-J)}(x)$ où J est le degré, et R est le tableau des coefficients de $Q_J^{(IA-J)}(t)$.

T, R, IA et J sont des paramètres d'entrée.

Description : Ce sous-programme calcule l'approximant de Padé $[IA-1/J]_f(t)$.

EXEMPLE D'APPEL DES SOUS-PROGRAMMES

On présente un programme qui lit les valeurs de NL et des coefficients de la série formelle, et fait appel aux sous-programmes.

```

1. C*****
2. C   PROGRAMME PRINCIPAL POUR LE CALCUL DES POLYNOMES ORTHOGONAUX ET LEURS
3. C   ASSOCIES-LA TABLE DES APPROXIMANTS DE PADE AUSSI
4. C   IL UTILISE APADÉ POUR LE CALCUL DES POLYNOMES, CE VERNIER APPELLE
5. C   ECRIRE POUR LE CALCUL DES APPROXIMANTS DE PADE
6. C   DC EST LE TABLEAU DES COEFFICIENTS DE LA SERIE FORMELLE, IL EST
7. C   INDICE DE MON A NL
8. C   DC EST TRANSMIS AUX SOUS PROGRAMMES PAR LE COMMON
9. C*****
10. C*****CARTES POUVANT ETRE MODIFIEES PAR L'UTILISATEUR*****
11. C   COMMONDCR(-20:20,0:20),DC(-20:20),DBR(-20:20,0:20),DB(-20:20,0:20)
12. C   1,DE(-20:20,0:20),NDF(-20:20,0:20)/S/N(20),K(20),NR(20)
13. C   LGB=105
14. C*****FIN*****
15. C   READ(LGB,10)NL
16. C   READ(LGB,20)(DC(I),I=0,NL)
17. C   10 FORMAT(I3)
18. C   20 FORMAT(5*12.6)
19. C   CALL APADÉ(NL),STOP
20. C   END

```

ESSAI NUMERIQUE

Pour $NL = 10$ et les c_i suivants :

```

C( 0)= .1000000000000000+01
C( 1)= -.1000000000000000+01
C( 2)= .0000000000000000+00
C( 3)= .0000000000000000+00
C( 4)= .0000000000000000+00
C( 5)= .0000000000000000+00
C( 6)= .1000000000000000+01
C( 7)= .0000000000000000+00
C( 8)= .0000000000000000+00
C( 9)= .0000000000000000+00
C(10)= .0000000000000000+00

```

LES RESULTATS SONT :

Le tableau suivant:



Donnons deux polynômes orthogonaux réguliers de la table P, leurs associés et les approximants de Padé correspondants :

Le polynôme $P_k^{(n)}$ donné sous la forme

$$a_0 X^{**} 0$$

$$a_1 X^{**} 1$$

$$a_K X^{**} K$$

où les a_i sont réels

est égal à $\sum_{s=0}^K a_s x^s$.

$$Q_5^{(0)}(x)$$

$$P_5^{(0)}(t)$$

.1000000000000000+01x** 0	.1000000000000000+01t** 0
.0000000000000000+00x** 1	.1000000000000000-01t** 1
.0000000000000000+00x** 2	.0000000000000000+00t** 2
-.1000000000000000+01x** 3	.0000000000000000+00t** 3
.1000000000000000+01x** 4	.0000000000000000+00t** 4
.0000000000000000+00x** 5	.1000000000000000+01t** 5

APPROXIMANT DE PADÉ (4, 5)

NUMERATEUR

DENOMINATEUR

.1000000000000000+01t** 0	.1000000000000000+01t** 0
-.1000000000000000+01t** 1	.0000000000000000+00t** 1
.0000000000000000+00t** 2	.0000000000000000+00t** 2
.0000000000000000+00t** 3	.0000000000000000+00t** 3
.1000000000000000+01t** 4	.1000000000000000+01t** 4
	.1000000000000000+01t** 5

$$Q_5^{(1)}(x)$$

$$P_5^{(1)}(t)$$

.0000000000000000+00X** 0	.1000000000000000+01T** 0
.0000000000000000+00X** 1	.0000000000000000+00T** 1
.0000000000000000+00X** 2	.0000000000000000+00T** 2
.0000000000000000+00X** 3	.0000000000000000+00T** 3
-.1000000000000000+01X** 4	.0000000000000000+00T** 4
.0000000000000000+00X** 5	.1000000000000000+01T** 5

APPROXIMANT DE PAGE 5. >>

NUMERATEUR

DENOMINATEUR

.1000000000000000+01T** 0	.1000000000000000+01T** 0
-.1000000000000000+01T** 1	.0000000000000000+00T** 1
.0000000000000000+00T** 2	.0000000000000000+00T** 2
.0000000000000000+00T** 3	.0000000000000000+00T** 3
.0000000000000000+00T** 4	.0000000000000000+00T** 4
.1000000000000000+01T** 5	.1000000000000000+01T** 5




```

1. SUBROUTINE APAGE(NL)
2. C*****
3. C CE SOUS-PROGRAMME CALCULE LES POLYNOMES ORTHOGONAUX REGULIERS ET
4. C LEURS ASSOCIES
5. C ET UTILISE LES SOUS PROGRAMMES QUDE , EST ET DETECT POUR LE CALCUL
6. C DE LA TABLE Q-U
7. C IL APPELLE ECRIRE POUR LE CALCUL DES APPROXIMANTS DE PADE
8. C DC EST LE TABLEAU DES COEFFICIENTS DE LA SERIE FORMELLE, IL EST INDICE
9. C DE NON A NL, NON ETANT LE NUMERO DE LA PREMIERE DIAGONALE DES TABLES
10. C DE POLYNOMES
11. C*****
12. C*****CARTES POUVANT ETRE MODIFIEES PAR L'UTILISATEUR*****
13. COMMON/DCK(-20:20,0:20),C(-20:20),DCK(-20:20,0:20),DC(-20:20,0:20),
14. 1DE(-20:20,0:20),NDF(-20:20,0:20)/S/N(20),K(20),NR(20)
15. DIMENSION C(20),W(0:20),R(0:20),P(0:20,0:20),PR(0:20,0:20)
16. DIMENSION NR(0:20,0:20),Q(0:20,0:20)
17. MAT=108
18. C*****FIN*****
19. IF(NL.LE.0)WRITE(MAT,1020);STOP
20. 1020 FORMAT(2X,'ERREUR NL DOIT ETRE STRICT POSITIF')
21. C*****
22. C APPEL DU SOUS PROGRAMME QUDE
23. C*****
24. CALL QUDE(NL,0,0,LA);IA=0;ML=(NL+1)/2;P(IA,0)=1;QR(IA,0)=1
25. Q(IA,0)=0;PR(IA,0)=0;IA=1
26. C*****
27. C
28. C CALCUL DES POLYNOMES ORTHOGONAUX REGULIERS DE LA DEMI-TABLE SUPERIEURE
29. C ET LEURS ASSOCIES
30. C
31. C*****
32. 100 IF(IA.GT.ML)IA=IA-1;GO TO 390
33. PR(IA,0)=0;QR(IA,0)=1;P(IA,0)=1;Q(IA,0)=0;J=1
34. C*****
35. C TEST SUR LA NON-REGULARITE DU POLYNOME
36. C*****
37. 340 KA=IA-J;IF(NDF(IA-J,J).EQ.0)WRITE(MAT,350)KA,J;GO TO 101
38. IF(NDF(IA-J+1,J-1).EQ.0)GO TO 102
39. DO 9 I=0,J-1
40. Y R(I)=Q(IA-J+1,I)-UG(IA-J,J)*Q(IA-J,I)+C(IA-J)*P(IA-J+1,I)
41. T(0)=-DC(IA-J,J)-P(IA-J,0);T(J)=1;R(J)=0;IF(J.EQ.1)GO TO 120
42. DO 2 I=1,J-1
43. 2 T(I)=P(IA-J+1,I-1)-DC(IA-J,J)*P(IA-J,I);GO TO 120
44. 350 FORMAT(1,3X,'LE POLYNOME('',I2,',',',I2,',') EST NON O.R.',/)
45. C*****
46. C CAS OU LE POLYNOME EST SUR L'EST D'UN BLOC
47. C*****
48. 102 CALL DETECT(IA-J,1,J,LA,L);NR1=NR(I);N1=N(L)
49. IS=IA-J-N1+NR1+1;IF(IA-J.GT.0)GO TO 103
50. C CAS OU LE BLOC EST TRONQUE DANS LE COIN NORD-EST
51. DO 6 I=0,IS
52. 6 W(IS-I)=UBR(N1-NR1-1+I,J);W(IS+1)=1
53. DO 30 I=0,J-1
54. T(I)=0;R(I)=0;M2=MIN(IS+1,I);M1=MAX(0,I+1-J)
55. DO 30 L=M1,M2
56. R(I)=R(I)+2(IA-J,I-L)*W(L)
57. 30 T(I)=T(I)+P(IA-J,I-L)*W(L)
58. DO 16 I=0,J-1
59. T(I)=T(I)+DCR(IA-J,J)*PR(IA-J,I);R(I)=R(I)+DCR(IA-J,J)*QR(IA-J,I)
60. 16 CONTINUE;T(J)=1;R(J)=0;GO TO 120
61. C*****
62. C CALCUL DU POLYNOME DE L'EST D'UN BLOC ET SON ASSOCIE
63. C*****
64. 103 A=DE(IA-J-1,J);IF(IA-J.LE.N1-NR1-1)A=-DE(IA-J-1,J)
65. DO 4 I=0,J-1
66. 4 T(I)=P(IA-J-1,I)+A*P(IA-J,I);T(J)=1
67. R(0)=-C(IA-J-1)*P(IA-J-1,0)+A*Q(IA-J,0);R(J)=0;IF(J.EQ.1)GO TO 120
68. DO 5 I=1,J-1
69. 5 R(I)=Q(IA-J-1,I-1)-C(IA-J-1)*P(IA-J-1,I)+A*Q(IA-J,I)
70. 120 WRITE(108,300)KA,J
71. DO 8 I=0,J
72. QR(IA-J,I)=Q(IA-J,I);PR(IA-J,I)=P(IA-J,I)
73. Q(IA-J,I)=R(I);P(IA-J,I)=T(I)
74. 8 WRITE(MAT,320)R(I),I,T(I),I
75. C APPEL A ECRIRE POUR LE CALCUL DE L'APPROXIMANT DE PADE
76. CALL ECRIRE(R,T,IA,J);GO TO 305
77. 101 P(IA-J,J)=Q;PR(IA-J,J)=0;Q(IA-J,J)=0;QR(IA-J,J)=0
78. 305 IF(J.LT.IA)=J+1;GO TO 340
79. IA=IA+1;GO TO 100
80. 320 FORMAT(10X,622,15,'X**',I2,3X,622,15,'T**',I2)
81. 300 FORMAT(1,1X,'('',I2,',',',I2,',')',17X,'ASSOCIE',13X,'POLYNOME O.R.',/)
82. C*****
83. C
84. C CALCUL DES POLYNOMES ORTHOGONAUX REGULIERS DE LA DEMI-TABLE INFERIEURE
85. C ET LEURS ASSOCIES
86. C
87. C*****

```

BUS
LILLE

```

88. 390 IF(MOD(NL,2).NE.0)NL=NL-1
89. 360 IA=IA+1;IF(IA.GT.NL)RETURN
90. P(IA,0)=1;PR(IA,0)=0;QR(IA,0)=1;Q(IA,0)=0;J=1
91. C*****
92. C TEST SUR LA NON-REGULARITE DU POLYNOME
93. C*****
94. 370 KA=IA-J;IF(NDF(IA-J,J).EQ.0)WRITE(MAT,350)KA,J;GO TO 201
95. IF(NDF(IA-J+1,J-1).EQ.0)GO TO 202
96. DO 29 I=0,J-1
97. 29 R(I)=Q(IA-J+1,I)-DQ(IA-J,J)*Q(IA-J,I)+C(IA-J)*P(IA-J+1,I)
98. T(0)=-DQ(IA-J,J)*P(IA-J,0);T(J)=1;R(J)=0;IF(J.EQ.1)GO TO 220
99. DO 22 I=1,J-1
100. 22 T(I)=P(IA-J+1,I-1)-DQ(IA-J,J)*P(IA-J,I);GO TO 220
101. C*****
102. C CAS OU LE POLYNOME EST SUR L'EST D'UN BLOC
103. C*****
104. 202 CALL DETECT(IA-J,1,J,LA,L);NR1=NR(L);N1=N(L)
105. A=DE(IA-J-1,J);IF(IA-J.LE.N1-NR1-1)A=-DE(IA-J-1,J)
106. DO 24 I=0,J-1
107. 24 T(I)=P(IA-J-1,I)+A*P(IA-J,I);T(J)=1
108. R(0)=-C(IA-J-1,0)+P(IA-J-1,0)+A*Q(IA-J,0);R(J)=0;IF(J.EQ.1)GO TO 220
109. DO 25 I=1,J-1
110. 25 R(I)=Q(IA-J-1,I-1)-C(IA-J-1)*P(IA-J-1,I)+A*Q(IA-J,I)
111. 220 WRITE(MAT,300)KA,J
112. DO 28 I=0,J
113. QR(IA-J,I)=Q(IA-J,I);PR(IA-J,I)=P(IA-J,I)
114. Q(IA-J,I)=R(I);P(IA-J,I)=T(I)
115. 28 WRITE(MAT,320)R(I),T(I),I
116. C*****
117. C APPEL A ECRIRE POUR LE CALCUL DE L'APPROXIMANT DE PADE
118. C*****
119. CALL ECRIRE(R,T,IA,J);GO TO 804
120. 201 P(IA-J,J)=0;PR(IA-J,J)=0;Q(IA-J,J)=0;QR(IA-J,J)=0
121. 804 IF(J.LT.NL)J=J+1;GO TO 370
122. NL=NL-1;GO TO 360
123. END

```

```

1. SUBROUTINE ECRIRE(R,T,IA,J)
2. C*****
3. C CE SOUS-PROGRAMME CALCULE L'APPROXIMANT DE PADE A PARTIR D'UN
4. C POLYNOME -T-, DE LA DIAGONALE IA-J ET DE DEGRE J, ET SON ASSOCIE -R-
5. C*****
6. C*****CARTES POUVANT ETRE MODIFIEES PAR L'UTILISATEUR*****
7. COMMON/CK(-20:20,U:20),C(-20:20)
8. DIMENSION T(0:J),R(0:J),W(0:20),X(0:20)
9. MAT=108
10. C**********FIN*****
11. DO 4 I=0,J-1
12. 4 W(I)=R(J-1-I);IB=IA-1;WRITE(MAT,300)IB,J;WRITE(MAT,202)
13. DO 5 I=1,J
14. R(I)=T(J-I);X(J-I)=W(J-I)
15. 5 CONTINUE;R(0)=T(J);IF(IA-J.EQ.0)GO TO 8
16. DO 6 I=0,IA-1
17. X(I)=0;M2=MIN(I,J);M1=MAX(0,I+J+1-IA);IF(I.GE. IA-J)X(I)=W(I-IA+J)
18. DO 6 L=M1,M2
19. 6 X(I)=X(I)+C(I-L)*R(L)
20. DO 1 I=0,J-1
21. 1 WRITE(MAT,200)X(I),I,R(I),I
22. IF(IA-1.LT.J)WRITE(MAT,201)R(J),J;RETURN
23. WRITE(MAT,200)X(J),J,R(J),J;IF(IA-1.LE.J)RETURN
24. DO 2 I=J+1,IA-1
25. 2 WRITE(MAT,205)X(I),I;RETURN
26. 300 FORMAT(//,1X,'APPROXIMANT DE PADE(',I2,',',I2,')',/)
27. 202 FORMAT(4A,'NUMERATEUR',19X,'DENOMINATEUR',/)
28. 200 FORMAT(3X,D22.15,'T+',I2,3X,D22.15,'T+',I2)
29. 201 FORMAT(3X,D22.15,'T+',I2)
30. 203 FORMAT(3X,D22.15,'T+',I2)
31. END

```

QUATRIEME PARTIE

FONCTIONNELLES LINEAIRES LACUNAIRES D'ORDRE $s+1$
I. DEFINITION ET RAPPELS

On appelle fonctionnelle linéaire lacunaire d'ordre $s+1$ une fonctionnelle linéaire $c^{(r)}$ telle que $c_{i+r} = 0$ pour $i \in \mathbb{N}$, $r+i \not\equiv 0 \pmod{s+1}$.

Nous allons proposer un algorithme de calcul des polynômes orthogonaux réguliers lacunaires et leurs associés, et de la table des approximants de Padé, à partir des coefficients $c_{n(s+1)}$ avec s un entier positif quelconque.

Donnons quelques résultats de [1, chapitre 3] qui vont servir dans toute la suite.

On peut toujours définir des fonctionnelles linéaires $u^{(n)}$ telles que

$$\forall n \in \mathbb{N} \quad u^{(n)}(x^i) = c_{(n+i)(s+1)}$$

où les $c_{(n+i)(s+1)}$ sont les coefficients d'une fonctionnelle linéaire lacunaire c d'ordre $s+1$.

On appellera table U (resp. P) la table des polynômes orthogonaux unitaires $U_k^{(n)}(x)$ (resp. $P_k^{(n)}(x)$) par rapport aux fonctionnelles linéaires $u^{(n)}$ (resp. $c^{(n)}$).

Théorème 4.2. [1, page 214] :

Si le polynôme $U_k^{(n+1)}(x)$ est orthogonal régulier et est au Nord-Ouest d'un bloc U de largeur a dans la table U , alors dans P le polynôme $P_{k(s+1)}^{n(s+1)+1}(x) = U_k^{(n+1)}(x^{s+1})$ est orthogonal régulier et est au Nord-Ouest d'un bloc P de largeur $(a+1)(s+1)-1$.

En notant $v_k^{(n)}(x)$ (resp. $Q_k^{(n)}(x)$) les polynômes associés aux polynômes orthogonaux $U_k^{(n)}(x)$ (resp. $P_k^{(n)}(x)$), on a les deux corollaires suivants :

Corollaire 4.3. : Si $U_k^{(n+1)}(x)$ est orthogonal régulier alors

$$P_{k(s+1)}^{n(s+1)+1}(x) = U_k^{(n+1)}(x^{s+1}) \quad (1)$$

et

$$Q_{k(s+1)}^{n(s+1)+1}(x) = v_k^{(n+1)}(x^{s+1}) \quad (2)$$

Démonstration : (1) est le résultat du corollaire 3.4. de [1] et (2) se démontre facilement à partir de (1). $\square$

Corollaire 4.4. : Si $U_k^{(n+1)}(x)$ est orthogonal régulier on a :

1° Si $U_{k+1}^{(n)}$ est orthogonal régulier

$$(3) \quad \begin{cases} P_{k(s+1)+1+i}^{n(s+1)-i}(x) = x^{i+1} P_{k(s+1)}^{n(s+1)+1}(x) \\ Q_{k(s+1)+1+i}^{n(s+1)-i}(t) = Q_{k(s+1)}^{n(s+1)+1}(t) \end{cases} \quad \text{pour } i \in \mathbb{N}, 0 \leq i \leq s-1$$

2° Si $U_k^{(n+2)}(x)$ est orthogonal régulier

$$(4) \quad \begin{cases} P_{k(s+1)}^{n(s+1)+2+i}(x) = P_{k(s+1)}^{n(s+1)+1}(x) \\ Q_{k(s+1)}^{n(s+1)+2+i}(t) = t^{1+i} Q_{k(s+1)}^{n(s+1)+1}(t) \end{cases} \quad \text{pour } i \in \mathbb{N}, 0 \leq i \leq s-1$$

Démonstration : Ce corollaire découle de nos théorème 4.2 et corollaire 4.3 et, des propriétés 2.4. et 2.6. de [1]. $\square$

Soit la série formelle $f(t) = \sum_{i=0}^{\infty} c_i t^i$ telle que $c_n = 0$ pour $n \not\equiv 0 \pmod{s+1}$.

Posons $f^*(t) = \sum_{i=0}^{\infty} c_{i(s+1)} t^i$ et définissons f_n par

$$f_n(t) = \sum_{i=0}^{\infty} c_{n+i} t^i.$$

Des relations entre les polynômes $U_k^{(n)}$ et $V_k^{(n)}$, et les polynômes $P_k^{(n)}$ et $Q_k^{(n)}$ on déduit la propriété suivante :

Propriété 4.5. : Si $U_k^{(n+1)}$ existe alors

$$[k-1/k]_{f_{n+1}^*}(t^{s+1}) = [k(s+1)-1/k(s+1)]_{f_{n(s+1)+1}}(t)$$

et

$$[n+k/k]_{f^*}(t^{s+1}) = [(n+k)(s+1)/k(s+1)]_f(t).$$

Remarque 4.6. : Les polynômes orthogonaux réguliers $U_k^{(n)}(x)$ et leurs associés $V_k^{(n)}(x)$ se calculent suivant l'algorithme de la partie III, on va les supposer connus.

Nous proposons un algorithme qui détermine les polynômes orthogonaux réguliers lacunaires $P_{k(s+1)}^{n(s+1)+1+i}(x)$ pour $0 \leq i \leq s$, $P_{k(s+1)+j}^{n(s+1)+1-j}(x)$ pour $1 \leq j \leq s$ et leurs associés à partir des polynômes $U_k^{(n+1)}(x)$ et $V_k^{(n+1)}(x)$.

Algorithme de passage des polynômes $U_k^{(n)}$ et $V_k^{(n)}$ aux polynômes $P_j^{(m)}$ et $Q_j^{(m)}$

Le passage d'un numéro au suivant se fait en séquence, sauf s'il y a un ordre aller en.

Soit L le plus grand degré des polynômes $U_k^{(n)}(x)$ de la ligne $n+1$.

On pose $k = 0$

1 - Si $n+1-k = 0$ fin

Si $U_k^{(n+1-k)}$ est non O.R. aller en 4.

alors $P_{k(s+1)}^{(n-k)(s+1)+1}$ est O.R. et,

$$P_{k(s+1)}^{(n-k)(s+1)+1}(x) = U_k^{(n+1-k)}(x^{s+1})$$

et

$$Q_{k(s+1)}^{(n-k)(s+1)+1}(t) = V_k^{(n+1-k)}(t^{s+1})$$

2 - Si $U_{k+1}^{(n-k)}$ est non O.R, c'est-à-dire si $U_k^{(n-k+1)}$ est à l'Ouest d'un bloc,

aller en 3

$$\left\{ \begin{array}{l} P_{k(s+1)+1+i}^{(n-k)(s+1)-i} \text{ est O.R.} \\ P_{k(s+1)+1+i}^{(n-k)(s+1)-i}(x) = x^{1+i} P_{k(s+1)}^{(n-k)(s+1)+1}(x) \\ \text{et} \\ Q_{k(s+1)+1+i}^{(n-k)(s+1)-i}(t) = Q_{k(s+1)}^{(n-k)(s+1)+1}(t) \\ \text{pour } i \in \mathbb{N}, 0 \leq i \leq s-1 \end{array} \right.$$

3 - Si $U_k^{(n-k+2)}$ est non O.R, c'est-à-dire si $U_k^{(n+1-k)}$ est au Nord d'un bloc,

aller en 4

$$\left\{ \begin{array}{l} P_{k(s+1)}^{(n-k)(s+1)+2+i} \text{ est O.R.} \\ P_{k(s+1)}^{(n-k)(s+1)+2+i}(x) = P_{k(s+1)}^{(n-k)(s+1)+1}(x) \\ \text{et} \\ Q_{k(s+1)}^{(n-k)(s+1)+2+i}(t) = t^{1+i} Q_{k(s+1)}^{(n-k)(s+1)+1}(t) \\ \text{pour } i \in \mathbb{N}, 0 \leq i \leq s-1 \end{array} \right.$$

4 - Si $k \geq L$ fin

$k+1 \rightarrow k$ et aller en 1.

II. MISE EN OEUVRE

La mise en oeuvre se fait avec les sous-programmes de calcul des éléments de la table q_d et, des sous-programmes LACUNA et IMLAC. Les polynômes $U_k^{(n)}$ et $V_k^{(n)}$ se calculent suivant l'algorithme de calcul des polynômes orthogonaux réguliers et leurs associés de III. Le passage des approximants de Padé de f^* à ceux de f se fait sans aucune difficulté, il en est de même pour celui des polynômes $U_k^{(n)}$ et $V_k^{(n)}$ aux polynômes $P_j^{(m)}$ et $Q_j^{(m)}$ qui se fait suivant l'algorithme que l'on vient de décrire.

L'essentiel des résultats est échangé entre le programme principal et les sous-programmes par l'intermédiaire de l'instruction COMMON. Dans le sous-programme LACUNA les polynômes $U_k^{(n)}(x)$, $V_k^{(n)}(x)$, $P_k^{(n)}(x)$ et $Q_k^{(n)}(x)$ sont notés de la même manière que les polynômes du sous-programme APADE de la partie III.

MODE D'EMPLOI DES SOUS-PROGRAMMES

* * * LACUNA * * * *

Appel : CALL LACUNA(L1, NS)

Paramètres : NS+1 est l'ordre de la fonctionnelle linéaire lacunaire. Le tableau des coefficients u_j est transmis au sous-programme LACUNA par COMMON, ce tableau est indicé de 0 à L1. NS et L1 sont des paramètres d'entrée.

Sous-programmes utilisés : QUDE, EST, DETECT et IMLAC.

Description : Calcul des polynômes orthogonaux réguliers lacunaires et leurs associés à partir des coefficients de la fonctionnelle linéaire lacunaire d'ordre NS+1 de la forme $c_n(N_{S+1})$. Ce sous-programme donne aussi l'allure de la table des polynômes orthogonaux lacunaires.

* * * * * IMLAC * * * * *

Appel : CALL IMLAC(T, R, IA, J, NS)

Paramètres : T est le tableau des coefficients du polynôme $U_J^{(IA-J)}(x)$ et R celui des coefficients de l'associé $V_J^{(IA-J)}(x)$, ces deux tableaux sont indicés de 0 à J.

NS est tel que NS+1 est l'ordre de la fonctionnelle linéaire lacunaire.

T, R, IA, J et NS sont des paramètres d'entrée.

Description : Ce sous-programme calcule l'approximant de Padé $[(IA-1)(NS+1)/J(NS+1)]_f$ de la série formelle lacunaire d'ordre NS+1

EXEMPLE D'APPEL DES SOUS-PROGRAMMES

On présente un programme qui lit L1, NS et les coefficients u_i , avant de faire appel à l'ensemble des sous-programmes.

```

1.  C*****
2.  C   PROGRAMME PRINCIPAL POUR LE CALCUL DES POLYNOMES ORTHOGONAUX
3.  C   LACUNAIRES, LEURS ASSOCIES ET LES APPROXIMANTS DE PADE
4.  C   NS EST L'ORDRE DE LA FONCTIONNELLE LACUNAIRE
5.  C   DC EST LE TABLEAU DES COEFFICIENTS DE LA FONCTIONNELLE LACUNAIRE
6.  C   D'INDICES DE LA FORME N*(NS+1)
7.  C   CE PROGRAMME APPELLE LACUNA POUR LE CALCUL DES POLYNOMES ET LEURS
8.  C   ASSOCIES
9.  C*****
10. C*****
11. C   CARTES POUVANT ETRE CHANGEES PAR L'UTILISATEUR
12. C   COMMONDCR(-20:20,0:20),DC(-20:20),DBR(-20:20,0:20),DC(-20:20,0:20)
13. C   1,DE(-20:20,0:20),NOF(-20:20,0:20)/S/NA(20),KA(20),NRA(20)
14. C   LGB=105
15. C*****+FIN*****
16. C   READ(LGB,10)N
17. C   READ(LGB,10)NS
18. C   READ(LGB,20)(DC(I),I=0,N)
19. C   10 FORMAT(I3)
20. C   20 FORMAT(5F10.4)
21. C   CALL LACUNA(N,NS) STOP
22. C   END

```


ESSAI NUMERIQUE

Pour $L1 = 5$, $NS = 2$ et les coefficients u_i suivants :

```

C( 0)= .0000000000000000+01
C( 1)= .6000000000000000+01
C( 2)= .0000000000000000+00
C( 3)= .1000000000000000+01
C( 4)= .1000000000000000+01
C( 5)= .4000000000000000+01

```

on a le tableau suivant :

N0= 1	
N0= 2	+
N0= 3	+
N0= 4	+
N0= 5	+
N0= 6	+
N0= 7	+
N0= 8	+
N0= 9	+
N0=10	+
N0=11	+
N0=12	+
N0=13	+
N0=14	+
N0=15	+
N0=16	+
N0=17	+
N0=18	+
N0=19	+

BUS
LILLE

où les numéros de gauche, les + et les 0 sont définis comme dans les résultats des essais numériques de III,

donne l'allure de la table P des polynômes orthogonaux lacunaires.

Donnons quelques polynômes orthogonaux réguliers lacunaires, leurs associés et l'approximant de Padé correspondant :

Les polynômes sont donnés sous la même forme que dans l'essai numérique de III.

$$Q_6^{(4)}(x)$$

.1000000000000000+01x**	0
.0000000000000000+00x**	1
.0000000000000000+00x**	2
.0000000000000000+00x**	3
.0000000000000000+00x**	4
.0000000000000000+00x**	5
.0000000000000000+00x**	6

$$P_6^{(4)}(t)$$

-.3000000000000000+01t**	0
.0000000000000000+00t**	1
.0000000000000000+00t**	2
-.1000000000000000+01t**	3
.0000000000000000+00t**	4
.0000000000000000+00t**	5
.1000000000000000+01t**	6

$$Q_7^{(3)}(x)$$

-.1700000000000000+02x**	0
.0000000000000000+00x**	1
.0000000000000000+00x**	2
-.6000000000000000+01x**	3
.0000000000000000+00x**	4
.0000000000000000+00x**	5
.6000000000000000+01x**	6

$$P_7^{(3)}(t)$$

.0000000000000000+00t**	0
-.3000000000000000+01t**	1
.0000000000000000+00t**	2
.0000000000000000+00t**	3
-.1000000000000000+01t**	4
.0000000000000000+00t**	5
.0000000000000000+00t**	6
.1000000000000000+01t**	7

$$Q_8^{(2)}(x)$$

-.1700000000000000+02x**	0
.0000000000000000+00x**	1
.0000000000000000+00x**	2
-.6000000000000000+01x**	3
.0000000000000000+00x**	4
.0000000000000000+00x**	5
.6000000000000000+01x**	6

$$P_8^{(2)}(t)$$

.0000000000000000+00t**	0
.0000000000000000+00t**	1
-.3000000000000000+01t**	2
.0000000000000000+00t**	3
.0000000000000000+00t**	4
-.1000000000000000+01t**	5
.0000000000000000+00t**	6
.0000000000000000+00t**	7
.1000000000000000+01t**	8

$$Q_6^{(5)}(x)$$

.0000000000000000+00x**	0
.1000000000000000+01x**	1
.0000000000000000+00x**	2
.0000000000000000+00x**	3
.0000000000000000+00x**	4
.0000000000000000+00x**	5
.0000000000000000+00x**	6

$$P_6^{(5)}(t)$$

-.3000000000000000+01t**	0
.0000000000000000+00t**	1
.0000000000000000+00t**	2
-.1000000000000000+01t**	3
.0000000000000000+00t**	4
.0000000000000000+00t**	5
.1000000000000000+01t**	6

$$Q_6^{(6)}(x)$$

.0000000000000000+00x**	0
.0000000000000000+00x**	1
.1000000000000000+01x**	2
.0000000000000000+00x**	3
.0000000000000000+00x**	4
.0000000000000000+00x**	5
.0000000000000000+00x**	6

$$P_6^{(6)}(t)$$

-.3000000000000000+01t**	0
.0000000000000000+00t**	1
.0000000000000000+00t**	2
-.1000000000000000+01t**	3
.0000000000000000+00t**	4
.0000000000000000+00t**	5
.1000000000000000+01t**	6

APPROXIMANT DE PADE(y , 6)

NUMERATEUR		DENOMINATEUR	
.0000000000000000+01X**	0	.1000000000000000+01T**	0
.0000000000000000+00X**	3	-.1000000000000000+01T**	3
-.2400000000000000+02X**	6	-.3000000000000000+01T**	6
-.1700000000000000+02X**	9		

CONCLUSION

Les résultats obtenus sont les mêmes que quand on utilise tous les coefficients de la fonctionnelle linéaire lacunaire.



LES SOUS-PROGRAMMES

```

1.      SUBROUTINE LACUNA(L1,NS)
2.      C*****
3.      C      CALCUL DES POLYNOMES ORTHOGONAUX LACUNAIRES A PARTIR DES COEFFICIENTS
4.      C      DE LA FONCTIONNELLE LINEAIRE D'INDICES DE LA FORME N*(NS+1). CES
5.      C      COEFFICIENTS SONT RANGES DANS LE TABLEAU-U-TRANSMIS AU SOUS-PRO
6.      C      GRAMME PAR COMMON
7.      C
8.      C*****
9.      C*****CARTES POUVANT ETRE CHANGEES PAR L'UTILISATEUR*****
10.     COMMONDCR(-20:20,0:20),U(-20:20),DBR(-20:20,0:20),DO(-20:20,0:20),
11.     1DE(-20:20,0:20),NMF(-20:20,0:20)/S/N(20),K(20),NR(20)
12.     DIMENSIONT(0:40),R(0:40),X(0:40),W(0:20),Y(0:40),NF(0:40,0:40)
13.     DIMENSIONP(0:20,0:20),PR(0:20,0:20),Q(0:20,0:20),QR(0:20,0:20)
14.     MAT=108
15.     C*****FIN*****
16.     CALL GUEDE(L1,0,0,LA);IA=0;ML=(L1+1)/2;J2=(L1+1)*(NS+1);I=0;J1=J2
17.     22 DO 21 J=0,J1+1
18.     21 NF(J,I)=0;IF(J1,LE,0)GO TO 170
19.     J1=J1-2;I=I+1;GO TO 22
20.     C*****
21.     C      CALCUL DES POLYNOMES -U- ET LEURS ASSOCIES -V-
22.     C*****
23.     170 P(IA,0)=1;PR(IA,0)=0;Q(IA,0)=0;QR(IA,0)=1;IA=1
24.     100 IF(IA,GT,ML)GO TO 390
25.     Q(IA,0)=1;QR(IA,0)=0;R(0)=0;P(IA,0)=0;PR(IA,0)=0;T(0)=1;KA=IA;J=0
26.     GO TO 120
27.     340 KA=IA-J;IF(NDF(IA-J,J),EQ,0)WRITE(MAT,350)KA,J;GO TO 101
28.     T(J)=1;R(J)=0;IF(NDF(IA-J+1,J-1),EQ,0)GO TO 102
29.     R(0)=Q(IA-J+1,0)-DO(IA-J,J)+Q(IA-J,0)+U(IA-J)*P(IA-J+1,0)
30.     T(0)=-DO(IA-J,J)+P(IA-J,0);IF(J,EQ,1)GO TO 120
31.     DO 2 I=1,J-1
32.     R(I)=Q(IA-J+1,I)-DO(IA-J,J)+Q(IA-J,I)+U(IA-J)*P(IA-J+1,I)
33.     2 T(I)=P(IA-J+1,I-1)-DO(IA-J,J)+P(IA-J,I);GO TO 120
34.     350 FORMAT(3A,'LE POLYNOME',I2,3X,I2,' EST NON O,R.')
35.     102 CALL DETECT(IA-J,1,J,LA,L);NR1=NR(L);N1=N(L)
36.     IS=IA-J-N1+NR1+1;IF(IA-J,GT,0)GO TO 103
37.     DO 6 I=0,IS
38.     6 W(IS-I)=DBR(N1-NR1-1+I,J);W(IS+1)=1.
39.     DO 30 I=0,J-1
40.     T(I)=0;R(I)=0;M2=MAX(0,I+1+IS-J);M3=MIN(1,IS+1)
41.     DO 30 L=M2,M3
42.     T(I)=T(I)+P(IA-I,I-L)*W(L);R(I)=R(I)+Q(IA-J,I-L)*W(L)
43.     30 CONTINUE
44.     DO 16 I=0,J-IS-2
45.     T(I)=T(I)+DCR(IA-J,J)+PR(IA-J,I);R(I)=R(I)+DCR(IA-J,J)+QR(IA-J,I)
46.     16 CONTINUE;GO TO 120
47.     103 A=DE(IA-J-1,J);IF(IA-J,LE,N1-NR1-1)A=-A
48.     R(0)=-U(IA-J-1)+P(IA-J,0)+A+Q(IA-J,0)
49.     DO 4 I=1,J-1
50.     R(I)=Q(IA-J-1,I-1)-U(IA-J-1)+P(IA-J-1,I)+A+Q(IA-J,I)
51.     4 T(I)=P(IA-J-1,I)+A+P(IA-J,I);T(0)=P(IA-J-1,0)+A+P(IA-J,0)
52.     1005 FORMAT(//)
53.     120 WRITE(MAT,800)KA,J
54.     DO 8 I=0,J
55.     QR(IA-J,I)=Q(IA-J,I);PR(IA-J,I)=P(IA-J,I)
56.     Q(IA-J,I)=R(I);P(IA-J,I)=T(I)
57.     8 WRITE(MAT,330)R(I),T(I),I;IF(IA-J,EQ,0)GO TO 601
58.     330 FORMAT(20X,D22,15,'X+',I3,3X,D22,15,'T+',I3)
59.     800 FORMAT(//,1X,'U(',I3,',',I3,')',18X,'ASSOCIE',13X,'POLYNOME',/)
60.     300 FORMAT(//,1X,'P(',I3,',',I3,')',18X,'ASSOCIE',13X,'POLYNOME',/)
61.     IF(J,GT,0)WRITE(MAT,1005)CALL IMLAC(T,R,IA,J,NS)
62.     C*****
63.     C      CALCUL DES POLYNOMES ORTHOGONAUX LACUNAIRES ET LEURS ASSOCIES
64.     C*****
65.     N2=(KA-1)*(NS+1)+1;K2=J*(NS+1);I2=0;WRITE(MAT,300)N2,K2
66.     DO 50 L=0,K2
67.     T(L)=0;R(L)=0
68.     IF(MOD(L,NS+1),EQ,0)T(L)=P(IA-J,I2);R(L)=Q(IA-J,I2);I2=I2+1
69.     50 WRITE(MAT,330)R(L),L,T(L),L,NF(N2,K2)=1
70.     IF(NDF(IA-J-1,J+1),EQ,0)GO TO 111
71.     DO 60 I=0,K2
72.     60 Y(I)=R(I)+U(IA-J-1)*T(I)
73.     DO 61 I=1,NS
74.     KA=N2-I;KJ=K2+I;IF(KA,LT,0)GO TO 111
75.     WRITE(MAT,1005);WRITE(MAT,300)KA,KJ
76.     DO 62 L=0,K2
77.     X(L)=0;IF(L,GE,I)X(L)=T(L-I)
78.     62 WRITE(MAT,330)Y(L),L,X(L),L,NF(KA,KJ)=1
79.     DO 64 J1=K2+1,K2+I
80.     X(J1)=0;IF(J1,GE,I)X(J1)=T(J1-I)
81.     64 WRITE(MAT,321)X(J1),J1
82.     61 CONTINUE
83.     321 FORMAT(51X,D22,15,'T+',I3)
84.     111 IF(NDF(IA-J+1,J),EQ,0)GO TO 601
85.     DO 65 L=1,NS
86.     WRITE(MAT,1005);KA=N2+L;WRITE(MAT,300)KA,K2,NF(KA,K2)=1
87.     DO 65 J1=0,K2
88.     X(J1)=0;IF(J1,GE,L)X(J1)=R(J1-L)

```



```

89.      65 WRITE(MAT,330)X(J1),J1,T(J1),J1;GO TO 601
90.      101 P(IA-J,J)=P(IA-J,J)+Q(IA-J,J)+Q(IA-J,J)+Q(IA-J,J)=0
91.      601 IF(J.LT.1A)J=J+1;GO TO 340
92.      1A=1A+1;GO TO 100
93.      390 IF(MOD(L1,2).NE.0) ML=ML-1
94.      360 IF(ML.LT.0)GO TO 400
95.      C*****
96.      C      CALCUL DES POLYNOMES -U- ET LEURS ASSOCIES -V-
97.      C*****
98.      Q(IA,0)=1;OR(IA,0)=0;R(0)=0;P(IA,0)=0;PR(IA,0)=0;T(0)=1;KA=IA;J=0
99.      GO TO 220
100.     370 KA=IA-J;IF(NDF(IA-J,J).EQ.0)WRITE(MAT,350)KA,J;GO TO 201
101.     T(J)=1;R(J)=0;IF(NDF(IA-J+1,J-1).EQ.0)GO TO 202
102.     R(0)=Q(IA-J+1,0)-DQ(IA-J,J)+Q(IA-J,0)+U(IA-J)*P(IA-J+1,0)
103.     T(0)=DQ(IA-J,J)+P(IA-J,0);IF(J.EQ.1)GO TO 220
104.     DO 32 I=1,J-1
105.     R(I)=Q(IA-J+1,I)-UQ(IA-J,J)+Q(IA-J,I)+U(IA-J)*P(IA-J+1,I)
106.     32 T(I)=P(IA-J+1,I-1)-DQ(IA-J,J)+P(IA-J,I);GO TO 220
107.     202 CALL DETECT(IA-J,1,J,LA,L);NR1=NR(L);N1=N(L)
108.     A=DE(IA-J-1,J);IF(IA-J.LE.N1-NR1-1)A=-A
109.     R(0)=U(IA-J-1)*P(IA-J,0)+A+Q(IA-J,0)
110.     DO 24 I=1,J-1
111.     R(I)=Q(IA-J-1,I-1)-U(IA-J-1)*P(IA-J-1,I)+A+Q(IA-J,I)
112.     24 T(I)=P(IA-J-1,I)+A+P(IA-J,I);T(n)=P(IA-J-1,0)+A+P(IA-J,0)
113.     220 WRITE(MAT,800)KA,J
114.     DO 28 I=0,J
115.     QR(IA-J,I)=Q(IA-J,I);PR(IA-J,I)=P(IA-J,I)
116.     Q(IA-J,I)=R(I);P(IA-J,I)=T(I)
117.     28 WRITE(MAT,330)R(I),I,T(I),I
118.     IF(J.GT.0)WRITE(MAT,1005);CALL IMLAC(T,R,IA,J,NS)
119.     C*****
120.     C      CALCUL DES POLYNOMES ORTHOGONAUX LACUNAIRES ET LEURS ASSOCIES
121.     C*****
122.     N2=(KA-1)*(NS+1)+1;K2=J*(NS+1);I2=0;WRITE(MAT,300)N2,K2
123.     DO 70 L=0,K2
124.     T(L)=0;R(L)=0
125.     IF(MOD(L,NS+1).EQ.0)T(L)=P(IA-J,I2);R(L)=Q(IA-J,I2);I2=I2+1
126.     70 WRITE(MAT,330)R(L),L,T(L),L,NF(N2,K2)=1
127.     IF(NDF(IA-J-1,J+1).EQ.0)GO TO 211
128.     DO 80 I=0,K2
129.     80 Y(I)=R(I)+U(IA-J-1)*T(I)
130.     DO 81 I=1,NS
131.     WRITE(MAT,1005);KA=N2-I;KJ=K2+I;WRITE(MAT,300)KA,KJ,NF(KA,KJ)=1
132.     DO 82 L=0,K2
133.     X(L)=0;IF(L.GE.I)X(L)=T(L-I)
134.     82 WRITE(MAT,330)Y(L),L,X(L),L
135.     DO 84 J1=K2+1,K2+I
136.     X(J1)=0;IF(J1.GE.I)X(J1)=T(J1-I)
137.     84 WRITE(MAT,321)X(J1),J1
138.     81 CONTINUE
139.     211 IF(NDF(IA-J+1,J).EQ.0)GO TO 901
140.     DO 85 L=1,NS
141.     WRITE(MAT,1005);KA=N2+L;WRITE(MAT,300)KA,K2,NF(KA,K2)=1
142.     DO 85 J1=0,K2
143.     X(J1)=0;IF(J1.GE.L)X(J1)=R(J1-L)
144.     85 WRITE(MAT,330)X(J1),J1,T(J1),J1;GO TO 901
145.     201 P(IA-J,J)=0;PR(IA-J,J)=0;Q(IA-J,J)=0;OR(IA-J,J)=0
146.     901 IF(J.LT.ML)J=J+1;GO TO 370
147.     1A=1A+1;ML=ML-1;GO TO 360
148.     C*****
149.     C      SCHEMATISATION DE LA TABLE DES POLYNOMES ORTHOGONAUX LACUNAIRES
150.     C*****
151.     400 I=0;M2=J2
152.     47 DO 45 J=1,M2
153.     IF(NF(J,I).NE.0)NF(J,I)=1H+;GO TO 45
154.     NF(J,I)=1H0
155.     45 CONTINUE;M2=M2-2;IF(M2.LE.0)ME=J2/2;GO TO 46
156.     I=I+1;GO TO 47
157.     46 KA=1;WRITE(MAT,805)KA
158.     DO 48 J=1,ME
159.     WRITE(MAT,1002);L=J+1
160.     48 WRITE(MAT,805)(L,(NF(J-I,I),I=0,J-1))
161.     DO 49 J=1,J2-ME;IF(ME+J.GT.J2)RETURN
162.     KA=ME+J+1;WRITE(MAT,1002)
163.     49 WRITE(MAT,805)(KA,(NF(ME+J-I,I),I=0,J2-ME-J+1))
164.     805 FORMAT(1X,'N0=',I2,5X,20(A1,5X))
165.     1002 FORMAT(/)
166.     END

```

```

1.      SUBROUTINE IMLAC(T,R,IA,J,NS)
2.      C*****
3.      C      CALCUL DE L'APPROXIMANT DE PADE D'UNE SERIE LACUNAIRE D'ORDRE NS
4.      C      T EST LE TABLEAU DES COEFFICIENTS DU POLYNOME -U- DE DEGRE J ET R EST
5.      C      LE TABLEAU DES COEFFICIENTS DE -V- .L'ASSOCIE DE -U-
6.      C*****
7.      C*****CARTES POUVANT ETRE CHANGEES PAR L'UTILISATEUR*****
8.      COMMONDCR(-20:20,0:20),U(-20:20)
9.      DIMENSIONT(0:J),R(0:J),X(0:40),Y(0:40)
10.     MAT=108
11.     C*****FIN*****
12.     KA=(IA-J-1)*(NS+1)+1;K2=J*(NS+1);JA=KA+K2-1;WRITE(MAT,200)JA,K2
13.     WRITE(MAT,300);I1=J;IF(JA.GT.0)GO TO 10
14.     DO 8 I=0,K2
15.     IF((MOD(I,NS+1).EQ.0).AND.(I.LT.K2))WRITE(MAT,5)R(I1-1),I,T(I1),I,
16.     1I1=I1-1;GO TO 8
17.     IF(I.EQ.K2)WRITE(MAT,7)T(0),K2
18.     CONTINUE;RETURN
19.     10 X(J)=0;IF(J.EQ.0)GO TO 3
20.     DO 1 I=0,J-1
21.     1 X(I)=R(J-1-I)
22.     3 DO 2 I=0,J
23.     2 R(I)=T(J-I)
24.     DO 4 I=0,IA-1
25.     Y(I)=0;M1=MAX(0,I-J);M2=MIN(I,IA-J-1);IF(I.GE. IA-J)Y(I)=X(I-IA+J)
26.     DO 4 L=M1,M2
27.     4 Y(I)=Y(I)+R(I-L)*U(L);I2=0
28.     300 FORMAT(9X,'NUMERATEUR',21X,'DENOMINATEUR')
29.     200 FORMAT(/,3X,'APPROXIMANT DE PADE(',I3,'.',I3,')'./)
30.     DO 6 I=0,KA+K2-1
31.     IF((I.LE.K2).AND.(MOD(I,NS+1).EQ.0))WRITE(MAT,5)Y(I2),I,R(I2),I,I2
32.     1=I2+1;GO TO 6
33.     IF(MOD(I,NS+1).EQ.0)WRITE(MAT,7)Y(I2),I;I2=I2+1
34.     6 CONTINUE;RETURN
35.     7 FORMAT(4X,D22.15,'x**',I3)
36.     5 FORMAT(4X,D22.15,'x**',I3,3X,D22.15,'t**',I3)
37.     END

```



CINQUIEME PARTIE

APPROXIMANTS DE PADE EN DEUX POINTS
I. DEFINITIONS ET RAPPELS

Soient deux séries formelles

$$L = \sum_{j=0}^{\infty} c_j x^j \quad \text{pour } x \text{ petit}$$

et

$$L^* = \sum_{j=1}^{\infty} c_{-j} x^{-j} \quad \text{pour } x \text{ grand}$$



On cherche la fraction rationnelle $[k/m]_{L,L^*}(x) = \frac{\sum_{j=0}^{m-1} a_j x^j}{\sum_{j=0}^m b_j x^j} = \frac{\tilde{Q}_{k,m}(x)}{\tilde{P}_{k,m}(x)}$

telle que $L \cdot \tilde{P}_{k,m}(x) - \tilde{Q}_{k,m}(x) = O_+(x^k)$ avec $0 \leq k \leq 2m$ (1)

et

$$L^* \cdot \tilde{P}_{k,m}(x) - \tilde{Q}_{k,m}(x) = O_-(x^{k-m-1}) \quad (2)$$

où $O_+(x^k)$ signifie que dans le second membre les coefficients de $x^0, x, \dots, x^{k-1}$ sont nuls

et

$O_-(x^{k-m-1})$ que dans le second membre les coefficients de $x^{k-m}, x^{k-m+1}, \dots, x^{m-1}$ sont nuls.

Définition 5.1. : $[k/m]_{L, L^*}(x)$ est appelée approximant de Padé en deux points L et L^* .

Remarque 5.2. : D'une manière générale si

$$L = \sum_{j=v}^{\infty} \bar{c}_j x^j \text{ pour } x \text{ petit} \quad \text{avec } v \in \mathbb{Z}$$

$$L^* = \sum_{j=-\infty}^{\mu} \bar{c}_j^* x^j \text{ pour } x \text{ grand} \quad \text{avec } \mu \in \mathbb{Z}$$

et $r = \max(0, -v)$,

l'approximant de Padé en deux points.

$$[k+l+r/m+r]_{L, L^*}(x) = \sum_{j=v}^{\ell-1} \bar{c}_j x^j + \sum_{j=\ell}^{\mu} \bar{c}_j^* x^j + x^{\ell} [k/m]_{L_{\ell}, L_{\ell}^*}(x)$$

avec $k \in \mathbb{N}$ tel que $v - \mu - 1 \leq k \leq 2m$

$\ell \in \mathbb{Z}$ tel que $\min(v, \mu+1) \leq \ell \leq \mu+1$

et la convention $\sum_n^{n'} \equiv 0$ pour $n' < n$.

$[k/m]_{L_{\ell}, L_{\ell}^*}(x)$ est l'approximant de Padé en deux points des séries

formelles

$$L_{\ell} = \sum_{j=0}^{\infty} (\bar{c}_{j+\ell} - \bar{c}_{j+\ell}^*) x^j \quad \text{pour } x \text{ petit}$$

et

$$L_{\ell}^* = \sum_{j=-\infty}^{-1} (\bar{c}_{j+\ell}^* - \bar{c}_{j+\ell}) x^j \quad \text{pour } x \text{ grand}$$

avec la convention $\bar{c}_j^* = 0$ si $j > \mu$ et $\bar{c}_j = 0$ si $j < v$.

On trouvera la définition 5.1. et la remarque 5.2. dans [4].

Remarque 5.3. : Dans toute la suite nous ne nous intéresserons qu'au calcul des approximants $[k/m]_{L_0, L_0^*}(x)$ des deux séries formelles

$$L_0 = \sum_{j=0}^{\infty} c_j x^j \quad \text{pour } x \text{ petit}$$

et

$$L_0^* = \sum_{j=1}^{\infty} c_{-j}^* x^{-j} \quad \text{pour } x \text{ grand}$$

Convention : $c_i = 0$ si $i < 0$ et $c_i^* = 0$ si $i > -1$

Des relations (1) et (2) on déduit le système linéaire de $2m$ équations à $2m+1$ inconnues suivant :

$$\begin{cases} \sum_{i=0}^m c_{j-i} b_i - a_j = 0 \text{ pour } j \in \mathbb{N}, 0 \leq j \leq k-1 \\ \sum_{i=0}^m c_{j-i} b_i - a_j = 0 \text{ pour } j \in \mathbb{N}, k-m \leq j \leq m-1 \end{cases}$$

avec la convention $a_j = 0$ si $j < 0$ ou $j > m-1$.

Posons $d_i = \bar{c}_i - \bar{c}_i^* \quad \forall i \in \mathbb{Z}$. Avec la convention sur c_i et c_i^* on peut

écrire $d_i = c_i - c_i^* \quad \forall i \in \mathbb{Z}$ ou encore

$$\begin{cases} d_i = c_i & \forall i \in \mathbb{N} \\ d_{-1-i} = -c_{-1-i}^* & \forall i \in \mathbb{N} \end{cases}$$

Définissons les fonctionnelles linéaires $d^{(j)}$ telles que

$$d^{(j)}(x^i) = d_{j+i} \quad \forall i \in \mathbb{N}, \forall j \in \mathbb{Z}.$$

Propriété 5.4. [1, page 445] : Si le polynôme $P_{k,m}$ existe, alors

$$P_{k,m}(x) = P_m^{(k-2m)}(x)$$

où $P_m^{(k-2m)}(x)$ est le polynôme unitaire de degré m orthogonal par rapport à la fonctionnelle linéaire $d^{(k-2m)}$.

On peut définir deux fonctionnelles linéaires, à partir des coefficients des séries formelles L_0 et L_0^* , telles que

$$c(x^i) = c_i \quad \forall i \in \mathbb{N}$$

et

$$c^*(x^i) = c_{-1-i}^* \quad \forall i \in \mathbb{N}.$$

Posons $\tilde{Q}_{k,m}(t) = t^{m-1} Q_{k,m}(t^{-1})$ et $\tilde{P}_{k,m}(x) = x^m P_{k,m}(x^{-1})$.

Propriété 5.5. [1, page 447] : Si $P_{k,m}(x)$ est orthogonal par rapport à $d^{(k-2m)}$ alors :

i) si $m-1 < k \leq 2m$ $Q_{k,m}(t) = c\left(\frac{P_{k,m}(x) - P_{k,m}(t)}{x - t}\right)$, la fonctionnelle c agissant sur la variable x .

ii) si $0 \leq k \leq m-1$ $\tilde{Q}_{k,m}(t) = c^*\left(\frac{\tilde{P}_{k,m}(x) - \tilde{P}_{k,m}(t)}{x - t}\right)$, la fonctionnelle c^* agissant sur la variable x .

Donnons le théorème 6.1. de [1, page 448] sur l'existence et l'unicité des approximants de Padé en deux points.

Théorème 5.6. : Soient $H_i^{(k-2m)}$ les déterminants de Hankel associés à la fonctionnelle linéaire $d^{(k-2m)}$.

i) Si $H_m^{(k-2m)} \neq 0$, l'approximant de Padé en deux points existe et est unique si on fixe le coefficient de x^m dans $P_{k,m}(x)$.

ii) Si $H_m^{(k-2m)} = 0$ et si h est le plus grand entier tel que $h+1 < m$ et $H_{h+1}^{(k-2m)} \neq 0$, et si p est le plus petit entier tel que $p+1 > m$ et $H_{p+1}^{(k-2m)} \neq 0$, alors :

a) Pour $h+2 \leq m \leq h + \text{entier}\left(\frac{p-h+1}{2}\right)$, l'approximant de Padé en deux points existe et est unique si on fixe le coefficient de x^m dans $P_{k,m}(x)$.

On a $[k/m]_{Lo, Lo^*}(t) = \frac{\gamma_{h+1}^{(k-2m)}(t)}{\gamma_{h+1}^{(k-2m)}(t)}$ avec $\gamma_{h+1}^{(k-2m)}(t) = t^{h+1} p_{h+1}^{(k-2m)}(t^{-1})$,

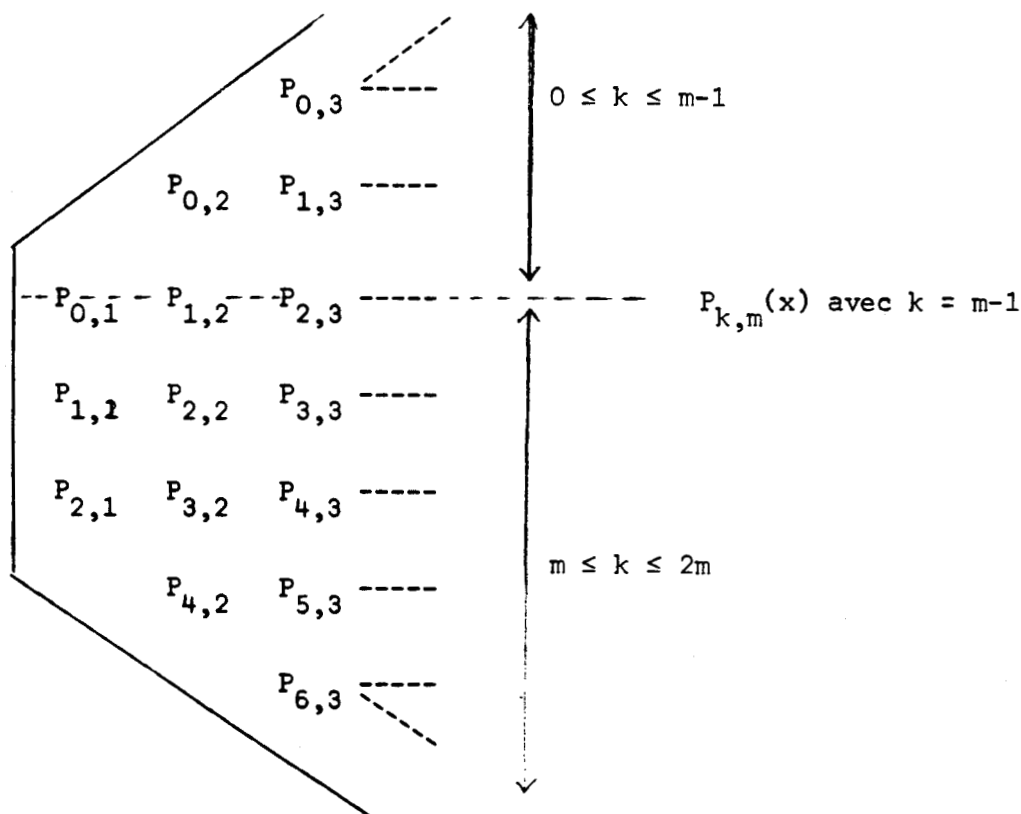
$p_{h+1}^{(k-2m)}(x)$ étant le polynôme de degré $h+1$ orthogonal régulier par rapport à la fonctionnelle linéaire $d^{(k-2m)}$,

et

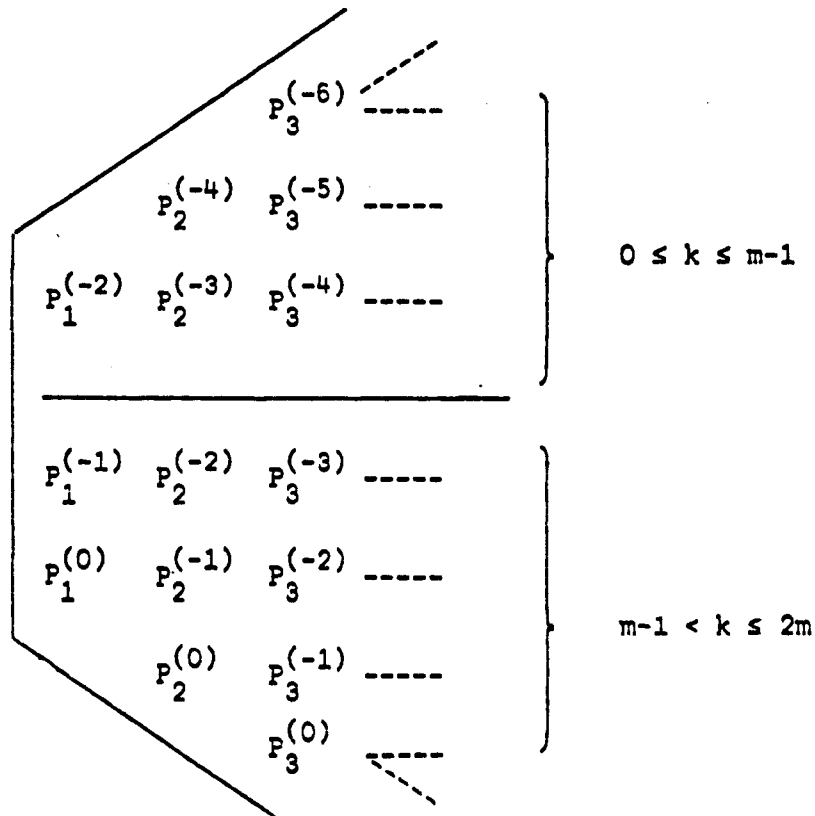
$$Q_{h+1}^{(k-2m)}(t) = \begin{cases} c^* \left(\frac{p_{h+1}^{(k-2m)}(x) - p_{h+1}^{(k-2m)}(t)}{x - t} \right) & \text{si } 0 \leq k \leq m-1 \\ t^h Q_{h+1}^{(k-2m)}(t^{-1}) & \text{si } m \leq k \leq 2m, \text{ avec} \\ Q_{h+1}^{(k-2m)}(t) = c \left(\frac{p_{h+1}^{(k-2m)}(x) - p_{h+1}^{(k-2m)}(t)}{x - t} \right) \end{cases}$$

b) Pour $h+1 + \text{entier}(\frac{p-h+1}{2}) \leq m \leq p$, l'approximant de Padé en deux points n'existe pas.

Définition 5.7. : On appellera table P_2 la table dans laquelle sont rangés les polynômes $P_{k,m}(x)$ de la manière suivante :



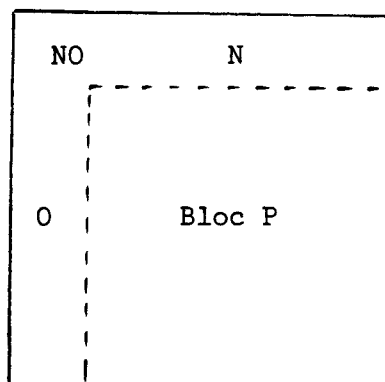
Cette table est donc une partie de la table P des polynômes orthogonaux par rapport aux fonctionnelles linéaires $d^{(j)}$, pour $j \in \mathbb{Z}$, $\mathbb{Z}$ étant l'ensemble des entiers relatifs.



Définition 5.8. : On appellera table Q_2 la table des polynômes $Q_{k,m}(x)$, elle a le même agencement que la table P_2 .

Définition 5.9. : On appellera table F_2 la table, qui a le même agencement que la table P_2 , dans laquelle sont rangés les approximants de Padé en deux points $[k/m]_{Lo,Lo^*}$.

On sait qu'un bloc F_2 est de la forme :



Propriété 5.10. [1, Corollaire 6.1, page 475] : Dans un bloc F_2 , tous les approximants de Padé en deux points situés au dessus de l'antidiagonale correspondant à l'antidiagonale principale du bloc P sont identiques lorsqu'on les a mis sous forme irréductible..

Donnons les relations de récurrence sur les diagonales et sur deux diagonales adjacentes qui vont nous permettre de calculer les polynômes $P_m^{(k-2m)}(x) = P_{k,m}(x)$ et leurs associés $Q_{k,m}(t)$.

Notations : Le polynôme orthogonal régulier qui précède le polynôme $P_i^{(n)}(x)$ sur la diagonale n étant noté $P_{pr(i,n)}^{(n)}(x)$ avec $pr(i,n)$ son degré, celui qui précède $P_{k,m}(x)$ sur la diagonale $k-2m$ sera noté $P_{k',pr(m,k-2m)}(x)$ avec k' tel que $k-2m = k'-2pr(m,k-2m)$. Le polynôme orthogonal régulier qui précède $P_{k',pr(m,k-2m)}(x)$ sur la diagonale $k-2m+1$ ou $k'-2pr(m,k-2m)+1$ sera noté $P_{k'',pr(pr(m,k-2m),k-2m+1)}(x)$ avec k'' tel que $k''-2pr(pr(m,k-2m),k-2m+1) = k'-2pr(m,k-2m)+1 = k-2m+1$.

Théorème 5.11. [1, page 477] : Les polynômes $P_{k,m}(x)$ de la table P_2 et leurs associés $Q_{k,m}(x)$ satisfont la même relation de récurrence à trois termes

$$U_{k,m}(x) = (x \omega_{m-1-pr(m,k-2m)}^{(k-2m)}(x) + B_m^{(k-2m)}) U_{k',pr(m,k-2m)}(x) + C_m^{(k-2m)} U_{k'',pr(pr(m,k-2m),k-2m)}(x)$$

avec k' et k'' tels que

$$k-2m = k'-2pr(m,k-2m) = k''-2pr(pr(m,k-2m),k-2m)$$

Propriété 5.12. [1, pages 496 et 500] :

- Si le polynôme $P_{k,m}(x)$ est orthogonal régulier alors $P_{k,m}(x)$ et $Q_{k,m}(x)$ vérifient la même relation de récurrence

$$U_{k,m}(x) = \omega_{m-pr(m+1,k-2m-1)}^{(k-2m-1)}(x) U_{k',pr(m+1,k-2m-1)}(x) + E_{m+1}^{(k-2m-1)} U_{k'',pr(pr(m+1,k-2m-1),k-2m)}(x)$$

avec k' et k'' tels que

$$k-2m = k'-2pr(m+1, k-2m-1)+1 = k''-2pr(pr(m+1, k-2m-1), k-2m)$$

et si de plus $P_{k+1, m+1}(x)$ est orthogonal régulier, $P_{k, m}(x)$ et $Q_{k, m}(x)$ vérifient la même relation de récurrence

$$x U_{k, m}(x) = U_{k+1, m+1}(x) + q_{m+1}^{(k-1-2m)} U_{k', pr(m+1, k-2m-1)}(x)$$

avec k' tel que $k-2m-1 = k'-2pr(m+1, k-2m-1)$

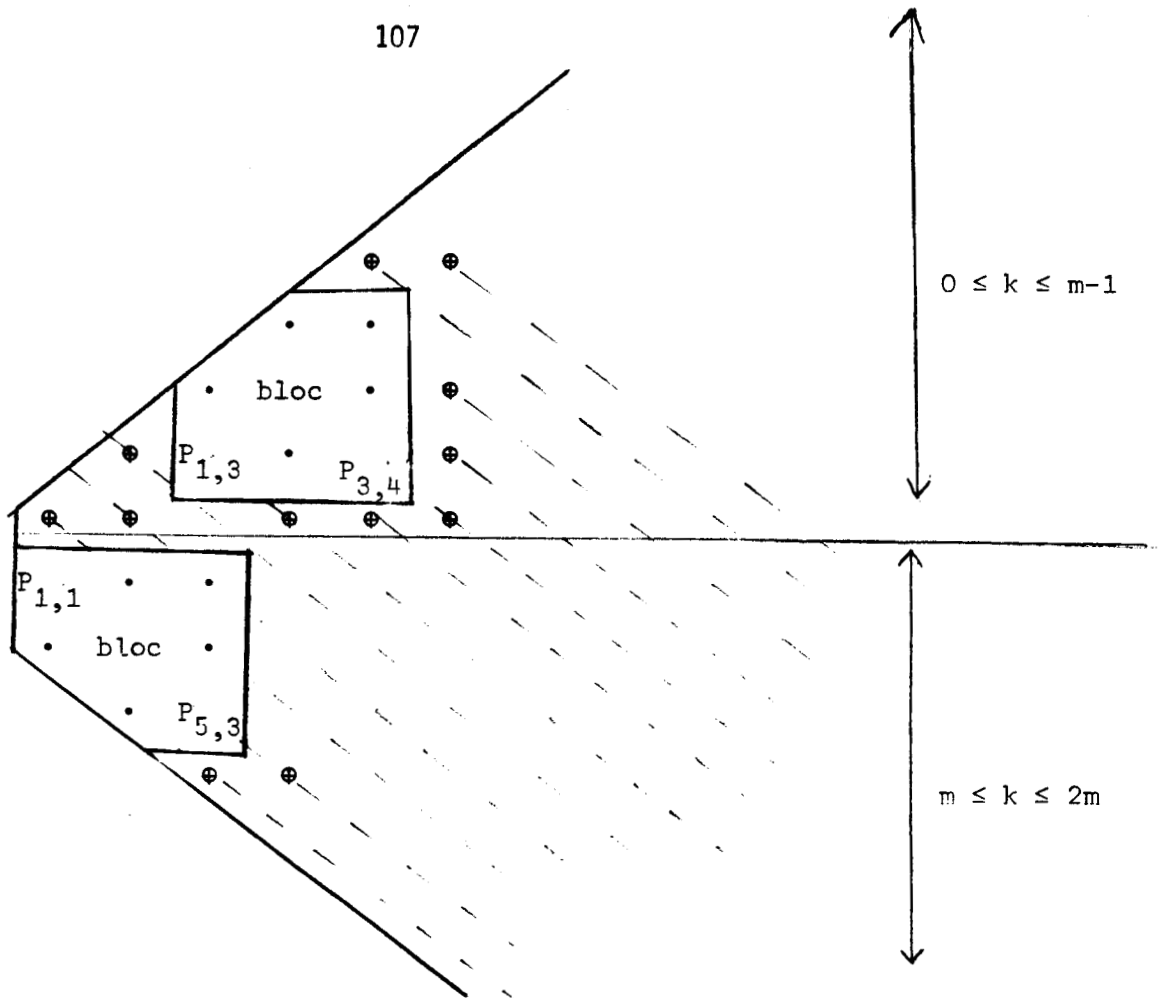
Remarque 5.13. : Les relations de récurrence sur les $Q_{k, m}(x)$ ne sont plus variables dès qu'un polynôme intervenant dans celles-ci n'est pas dans la table Q_2 .

II. MISE EN OEUVRE

NOTATION : On écrira O.R. pour dire orthogonal régulier.

Les polynômes $P_{k, m}(x)$ et $Q_{k, m}(x)$ se calculent suivant l'algorithme de calcul des polynômes O.R. de III. Si le calcul des $P_{k, m}(x)$ se fait sans aucune difficulté puisque la table P_2 est une partie de la table P des polynômes orthogonaux $P_m^{(k-2m)}(x)$, celui des $Q_{k, m}(x)$ nécessite la connaissance des associés des premiers polynômes orthogonaux réguliers de chaque diagonale pour démarrer les relations de la propriété 5.12, ces associés seront calculés à partir des relations de la propriété 5.5.

Exemple : Pour la table P_2 suivante :



les $\odot$ représentent les positions des polynômes $Q_{k,m}(x)$ que l'on doit connaître pour le démarrage des relations de récurrence de la propriété 5.12. Et donc il faut ajouter à l'algorithme de calcul des polynômes O.R. de III une séquence qui permet de savoir si les conditions d'utilisation de 5.12 sont satisfaites, sinon on calcule $Q_{k,m}(x)$ à l'aide de la propriété 5.5.

Remarque 5.14. : Dans le souci de simplification de la mise en oeuvre, on utilisera la propriété 5.5 si $Q_{k,m}(x)$ est sur la diagonale zéro ou si les conditions d'utilisation de la propriété 5.12 ne sont pas satisfaites.

Définition 5.15. : On appellera demi-table supérieure de la table P_2 (resp. Q_2) la partie contenant les polynômes $P_{k,m}(x)$ (resp. $Q_{k,m}(x)$) avec $0 \leq k \leq m-1$, le reste de la table sera appelé demi-table inférieure.

La mise en oeuvre s'effectue à l'aide des sous-programmes de calcul des éléments de la table qd et des sous-programmes PADPOI et IMPADE.

Dans le sous-programme PADPOI on notera :

- $P_{k,m}(x)$ par $P(K-2M, M)$
- $P_{k',pr(m,k-2m)}(x)$, avec k' tel que $k-2m = k'-2pr(m,k-2)$ par $PR(K-2M, M)$

et,

on fait de même pour $Q_{k,m}(x)$ avec Q à la place de P .

Les tableaux $P(K-2M,M)$, $PR(K-2M,M)$ et $Q(K-2M,M)$ sont définis comme dans III.

Les valeurs des éléments de la table qd , des coefficients d_i et des coefficients $C_k^{(n)}$ et $B_k^{(n)}$ intervenant dans les relations de récurrence à trois termes sur une diagonale sont échangés entre programme principal et sous-programmes par COMMON.

MODE D'EMPLOI DES SOUS-PROGRAMMES

* * * PADPOI * * * *

Appel : CALL PADPOI(C1, C2, N)

Paramètres : C_1 (resp. C_2) est le tableau des coefficients de la série formelle L (resp L^*), il est indicé de $-N$ à N .

C_1 , C_2 et N sont des paramètres d'entrée.

Sous-programmes utilisés : QUDE, EST, DETECT et IMPADE.

Description : Calcul des polynômes orthogonaux réguliers $P_{k,m}(x)$ et de leurs associés $Q_{k,m}(x)$.

* * * IMPADE * * * *

Appel : CALL IMPADE(X, Y, K, J)

Paramètres : X (resp. Y) est la table des coefficients du polynôme $P_{K,J}(x)$ (resp. $Q_{K,J}(x)$).

X , Y , K et J sont des paramètres d'entrée.

Description : Impression de l'approximant de Padé en deux points $[K/J]_{L_0, L_0^*}(x)$.

EXEMPLE D'APPEL DES SOUS-PROGRAMMES

On présente un programme qui lit les valeurs de N , des $\bar{c}_i$ et des $\bar{c}_i^*$, et fait appel à l'ensemble des sous-programmes.

```

1. C*****
2. C PROGRAMME PRINCIPAL POUR LE CALCUL DE LA TABLE DES APPROXIMANTS DE PADE
3. C EN DEUX POINTS
4. C T1 ET T2 SONT DES TABLEAUX INDICE DE -N A N
5. C T1 EST LE TABLEAU DES COEFFICIENTS DE LA SERIE QUE L'ON APPROXIME
6. C AU VOISINAGE DE ZERO
7. C T2 EST LE TABLEAU DES COEFFICIENTS DE LA SERIE QUE L'ON APPROXIME
8. C A L'INFINI
9. C*****
10. C*****CARTES POUVANT ETRE MODIFIEE PAR L'UTILISATEUR*****
11. COMMONDCR(-20:20,0:20),DC(-20:20),DBR(-20:20,0:20),DQ(-20:20,0:20)
12. 1,DE(-20:20,0:20),NDF(-20:20,0:20)/S/NA(20),KA(20),NRA(20)
13. DIMENSIONT1(-20:20),T2(-20:20)
14. MAT=108,LGB=105
15. C*****FIN*****
16. READ(LGB,1)N
17. 10 FORMAT(I5)
18. READ(LGB,20)(T1(I),I=-N,N)
19. READ(LGB,20)(T2(I),I=-N,N)
20. 20 FORMAT(5F15,8)
21. DO 3 I=-N,N
22. 3 WRITE(MAT,200)I,T1(I),I,T2(I),CALL PADPOI(T1,T2,N),STOP
23. 200 FORMAT(3X,'T1(',I3,')= ',D22.15,10X,'T2(',I3,')= ',D22.15)
24. END

```

ESSAI NUMERIQUE

Pour $N = 10$ et les $\bar{c}_i$ et $\bar{c}_i^*$ suivants :

T1(-10)=	.00000000000000000000	T2(-10)=	.00000000000000000000
T1(-9)=	.00000000000000000000	T2(-9)=	.00000000000000000000
T1(-8)=	.50000000000000000001	T2(-8)=	.10000000000000000001
T1(-7)=	.00000000000000000000	T2(-7)=	.00000000000000000000
T1(-6)=	.00000000000000000000	T2(-6)=	.00000000000000000000
T1(-5)=	.10000000000000000001	T2(-5)=	.80000000000000000001
T1(-4)=	.00000000000000000000	T2(-4)=	.00000000000000000000
T1(-3)=	.00000000000000000000	T2(-3)=	.00000000000000000000
T1(-2)=	.00000000000000000000	T2(-2)=	.00000000000000000000
T1(-1)=	.00000000000000000000	T2(-1)=	.00000000000000000000
T1(0)=	.00000000000000000000	T2(0)=	.00000000000000000000
T1(1)=	.10000000000000000001	T2(1)=	.90000000000000000001
T1(2)=	.00000000000000000000	T2(2)=	.00000000000000000000
T1(3)=	.00000000000000000000	T2(3)=	.00000000000000000000
T1(4)=	.10000000000000000001	T2(4)=	.20000000000000000001
T1(5)=	.00000000000000000000	T2(5)=	.00000000000000000000
T1(6)=	.00000000000000000000	T2(6)=	.00000000000000000000
T1(7)=	.10000000000000000001	T2(7)=	.15000000000000000002
T1(8)=	.00000000000000000000	T2(8)=	.00000000000000000000
T1(9)=	.00000000000000000000	T2(9)=	.00000000000000000000
T1(10)=	.10000000000000000001	T2(10)=	.30000000000000000001



T1 (resp. T2) est le tableau des coefficients $\bar{c}_i$ (resp. $\bar{c}_i^*$).

On a le tableau suivant :

The diagram illustrates a grid of points (marked with '+') and a central region (shaded gray) bounded by a diamond shape. The grid is labeled with 'N' values from -10 to 11 on the left and 'k' values from 0 to m-1 on the right. A vertical arrow on the right indicates the direction of increasing k.

The grid structure is as follows:

- Left side (N values):** N = -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11.
- Right side (k values):** $0 \leq k \leq m-1$ and $m-1 < k \leq 2m$.

The central region is shaded gray and contains the following points (marked with '+'):

- Top row (N = -4):** (0, -4), (1, -4), (2, -4), (3, -4), (4, -4), (5, -4), (6, -4), (7, -4), (8, -4), (9, -4), (10, -4), (11, -4), (12, -4), (13, -4), (14, -4), (15, -4), (16, -4), (17, -4), (18, -4), (19, -4), (20, -4), (21, -4), (22, -4), (23, -4), (24, -4), (25, -4), (26, -4), (27, -4), (28, -4), (29, -4), (30, -4), (31, -4), (32, -4), (33, -4), (34, -4), (35, -4), (36, -4), (37, -4), (38, -4), (39, -4), (40, -4), (41, -4), (42, -4), (43, -4), (44, -4), (45, -4), (46, -4), (47, -4), (48, -4), (49, -4), (50, -4), (51, -4), (52, -4), (53, -4), (54, -4), (55, -4), (56, -4), (57, -4), (58, -4), (59, -4), (60, -4), (61, -4), (62, -4), (63, -4), (64, -4), (65, -4), (66, -4), (67, -4), (68, -4), (69, -4), (70, -4), (71, -4), (72, -4), (73, -4), (74, -4), (75, -4), (76, -4), (77, -4), (78, -4), (79, -4), (80, -4), (81, -4), (82, -4), (83, -4), (84, -4), (85, -4), (86, -4), (87, -4), (88, -4), (89, -4), (90, -4), (91, -4), (92, -4), (93, -4), (94, -4), (95, -4), (96, -4), (97, -4), (98, -4), (99, -4), (100, -4), (101, -4), (102, -4), (103, -4), (104, -4), (105, -4), (106, -4), (107, -4), (108, -4), (109, -4), (110, -4), (111, -4), (112, -4), (113, -4), (114, -4), (115, -4), (116, -4), (117, -4), (118, -4), (119, -4), (120, -4), (121, -4), (122, -4), (123, -4), (124, -4), (125, -4), (126, -4), (127, -4), (128, -4), (129, -4), (130, -4), (131, -4), (132, -4), (133, -4), (134, -4), (135, -4), (136, -4), (137, -4), (138, -4), (139, -4), (140, -4), (141, -4), (142, -4), (143, -4), (144, -4), (145, -4), (146, -4), (147, -4), (148, -4), (149, -4), (150, -4), (151, -4), (152, -4), (153, -4), (154, -4), (155, -4), (156, -4), (157, -4), (158, -4), (159, -4), (160, -4), (161, -4), (162, -4), (163, -4), (164, -4), (165, -4), (166, -4), (167, -4), (168, -4), (169, -4), (170, -4), (171, -4), (172, -4), (173, -4), (174, -4), (175, -4), (176, -4), (177, -4), (178, -4), (179, -4), (180, -4), (181, -4), (182, -4), (183, -4), (184, -4), (185, -4), (186, -4), (187, -4), (188, -4), (189, -4), (190, -4), (191, -4), (192, -4), (193, -4), (194, -4), (195, -4), (196, -4), (197, -4), (198, -4), (199, -4), (200, -4), (201, -4), (202, -4), (203, -4), (204, -4), (205, -4), (206, -4), (207, -4), (208, -4), (209, -4), (210, -4), (211, -4), (212, -4), (213, -4), (214, -4), (215, -4), (216, -4), (217, -4), (218, -4), (219, -4), (220, -4), (221, -4), (222, -4), (223, -4), (224, -4), (225, -4), (226, -4), (227, -4), (228, -4), (229, -4), (230, -4), (231, -4), (232, -4), (233, -4), (234, -4), (235, -4), (236, -4), (237, -4), (238, -4), (239, -4), (240, -4), (241, -4), (242, -4), (243, -4), (244, -4), (245, -4), (246, -4), (247, -4), (248, -4), (249, -4), (250, -4), (251, -4), (252, -4), (253, -4), (254, -4), (255, -4), (256, -4), (257, -4), (258, -4), (259, -4), (260, -4), (261, -4), (262, -4), (263, -4), (264, -4), (265, -4), (266, -4), (267, -4), (268, -4), (269, -4), (270, -4), (271, -4), (272, -4), (273, -4), (274, -4), (275, -4), (276, -4), (277, -4), (278, -4), (279, -4), (280, -4), (281, -4), (282, -4), (283, -4), (284, -4), (285, -4), (286, -4), (287, -4), (288, -4), (289, -4), (290, -4), (291, -4), (292, -4), (293, -4), (294, -4), (295, -4), (296, -4), (297, -4), (298, -4), (299, -4), (300, -4), (301, -4), (302, -4), (303, -4), (304, -4), (305, -4), (306, -4), (307, -4), (308, -4), (309, -4), (310, -4), (311, -4), (312, -4), (313, -4), (314, -4), (315, -4), (316, -4), (317, -4), (318, -4), (319, -4), (320, -4), (321, -4), (322, -4), (323, -4), (324, -4), (325, -4), (326, -4), (327, -4), (328, -4), (329, -4), (330, -4), (331, -4), (332, -4), (333, -4), (334, -4), (335, -4), (336, -4), (337, -4), (338, -4), (339, -4), (340, -4), (341, -4), (342, -4), (343, -4), (344, -4), (345, -4), (346, -4), (347, -4), (348, -4), (349, -4), (350, -4), (351, -4), (352, -4), (353, -4), (354, -4), (355, -4), (356, -4), (357, -4), (358, -4), (359, -4), (360, -4), (361, -4), (362, -4), (363, -4), (364, -4), (365, -4), (366, -4), (367, -4), (368, -4), (369, -4), (370, -4), (371, -4), (372, -4), (373, -4), (374, -4), (375, -4), (376, -4), (377, -4), (378, -4), (379, -4), (380, -4), (381, -4), (382, -4), (383, -4), (384, -4), (385, -4), (386, -4), (387, -4), (388, -4), (389, -4), (390, -4), (391, -4), (392, -4), (393, -4), (394, -4), (395, -4), (396, -4), (397, -4), (398, -4), (399, -4), (400, -4), (401, -4), (402, -4), (403, -4), (404, -4), (405, -4), (406, -4), (407, -4), (408, -4), (409, -4), (410, -4), (411, -4), (412, -4), (413, -4), (414, -4), (415, -4), (416, -4), (417, -4), (418, -4), (419, -4), (420, -4), (421, -4), (422, -4), (423, -4), (424, -4), (425, -4), (426, -4), (427, -4), (428, -4), (429, -4), (430, -4), (431, -4), (432, -4), (433, -4), (434, -4), (435, -4), (436, -4), (437, -4), (438, -4), (439, -4), (440, -4), (441, -4), (442, -4), (443, -4), (444, -4), (445, -4), (446, -4), (447, -4), (448, -4), (449, -4), (450, -4), (451, -4), (452, -4), (453, -4), (454, -4), (455, -4), (456, -4), (457, -4), (458, -4), (459, -4), (460, -4), (461, -4), (462, -4), (463, -4), (464, -4), (465, -4), (466, -4), (467, -4), (468, -4), (469, -4), (470, -4), (471, -4), (472, -4), (473, -4), (474, -4), (475, -4), (476, -4), (477, -4), (478, -4), (479, -4), (480, -4), (

où les numéros de gauche sont les indices des diagonales de la table P, les + correspondent aux polynômes orthogonaux réguliers et les 0 à ceux qui sont dans les blocs.

qui donne l'allure de la table P_2 des polynômes $P_{k,m}$.

Donnons quatre approximants de Padé en deux points :

Les polynômes sont donnés sous la même forme que dans l'essai numérique de III.

APPROXIMANT DE PADE EN DEUX POINTS(5, 6)

NUMERATEUR		DENOMINATEUR	
.0000000000000000+00X**	0	.1000000000000000+01X**	0
-.8000000000000000+01X**	1	.0000000000000000+00X**	1
.0000000000000000+00X**	2	.0000000000000000+00X**	2
.0000000000000000+00X**	3	-.1250000000000000+00X**	3
.0000000000000000+00X**	4	.0000000000000000+00X**	4
.0000000000000000+00X**	5	.0000000000000000+00X**	5
		-.1142857142857140+01X**	6

APPROXIMANT DE PADE EN DEUX POINTS(6, 6)

NUMERATEUR		DENOMINATEUR	
.0000000000000000+00X**	0	.1000000000000000+01X**	0
-.8000000000000000+01X**	1	.0000000000000000+00X**	1
.0000000000000000+00X**	2	.0000000000000000+00X**	2
.0000000000000000+00X**	3	-.1250000000000000+00X**	3
.0000000000000000+00X**	4	.0000000000000000+00X**	4
.0000000000000000+00X**	5	.0000000000000000+00X**	5
		-.1142857142857140+01X**	6

APPROXIMANT DE PADE EN DEUX POINTS(9, 9)

NUMERATEUR		DENOMINATEUR	
.0000000000000000+00X**	0	.1000000000000000+01X**	0
-.8000000000000000+01X**	1	.0000000000000000+00X**	1
.0000000000000000+00X**	2	.0000000000000000+00X**	2
.0000000000000000+00X**	3	.9951690821256040+00X**	3
-.8961352657004830+01X**	4	.0000000000000000+00X**	4
.0000000000000000+00X**	5	.0000000000000000+00X**	5
.0000000000000000+00X**	6	-.1274396135265700+01X**	6
.0000000000000000+00X**	7	.0000000000000000+00X**	7
.0000000000000000+00X**	8	.0000000000000000+00X**	8
		-.1280193236714980+01X**	9

APPROXIMANT DE PADE EN DEUX POINTS(10, 9)

NUMERATEUR		DENOMINATEUR	
.0000000000000000+00X**	0	.1000000000000000+01X**	0
-.8000000000000000+01X**	1	.0000000000000000+00X**	1
.0000000000000000+00X**	2	.0000000000000000+00X**	2
.0000000000000000+00X**	3	.9951690821256040+00X**	3
-.8961352657004830+01X**	4	.0000000000000000+00X**	4
.0000000000000000+00X**	5	.0000000000000000+00X**	5
.0000000000000000+00X**	6	-.1274396135265700+01X**	6
.0000000000000000+00X**	7	.0000000000000000+00X**	7
.0000000000000000+00X**	8	.0000000000000000+00X**	8
		-.1280193236714980+01X**	9



Ces résultats montrent que l'on a bien identifié des approximaux de Padé en deux points $[k/m]_{L_0, L_0^*}(x)$ à l'ouest, au Nord et dans le coin Nord-Ouest d'un bloc P_2 .

LES SOUS-PROGRAMMES


```

1.      SUBROUTINE PADPOI(C1,C2,NL)
2.      C*****
3.      C      CALCUL DES POLYNOMES ORTHOGONAUX REGULIERS ET LEURS ASSOCIES EN VUE
4.      C      DU CALCUL DE LA TABLE DES APPROXIMANTS DE PADE EN DEUX POINTS
5.      C      C1 EST LE TABLEAU DES COEFFICIENTS DE LA SERIE QUE L'APPROXIME AU
6.      C      VOISINAGE DE ZERO
7.      C      C2 EST LE TABLEAU DES COEFFICIENTS DE LA SERIE QUE L'ON APPROXIME
8.      C      A L'INFINI
9.      C      CES DEUX TABLEAUX SONT INDICES DE -NL A NL
10.     C      CE SOUS PROGRAMME UTILISE LE SOUS PROGRAMME IMPADE POUR IMPRIMER
11.     C      LES APPROXIMANTS DE PADE
12.     C*****
13.     C*****CARTES POUVANT ETRE MODIFIEES PAR L'UTILISATEUR*****
14.     COMMON/DCR(-20:20,0:20),T1(-20:20),DBR(-20:20,0:20),DC(-20:20,0:20)
15.     --1,DE(-20:20,0:20),NDF(-20:20,0:20)/S/N(20),K(20),NR(20)
16.     DIMENSIONP(-20:20,0:20),PR(-20:20,0:20),W(0:20)
17.     DIMENSIONC1(-20:NL),C2(-20:NL),T(0:20),Q(-20:20,0:20),R(0:20)
18.     MAT=108
19.     C*****FIN*****
20.     IF(NL.LE.0)WRITE(MAT,500);STOP
21.     500 FORMAT(3X,'ERREUR LA VALEUR DE NL DOIT ETRE SUP STRICT A ZERO')
22.     DO 37 I=-NL,NL
23.     37 T1(I)=C1(I)-C2(I)
24.     C      APPEL DE QUDE
25.     CALL QUDECNL,-NL,0,LA;IA=-NL;ML=NL
26.     C*****
27.     C      CALCUL DES POLYNOMES ORTHOGONAUX REGULIERS DE LA DEMI-TABLE SUPERIEURE
28.     C      ET LEURS ASSOCIES
29.     C
30.     C
31.     C*****
32.     P(IA,0)=1;PR(IA,0)=0;Q(IA,0)=0;IA=IA+1
33.     100 IF(IA.GT.-1)GO TO 360
34.     P(IA,0)=1;PR(IA,0)=0;Q(IA,0)=0;J=1
35.     340 K2=IA+J;KA=IA-J;IF(NDF(IA-J,J).EQ.0)WRITE(MAT,350)KA,J;GO TO 101
36.     T(J)=1;R(J)=0;IF(NDF(IA-J+1,J-1).EQ.0)GO TO 102
37.     T(0)=-DQ(IA-J,J)*P(IA-J,0);R(0)=0;IF(J.EQ.1)GO TO 110
38.     DO 2 I=1,J-1
39.     R(I)=0
40.     2 T(I)=P(IA-J+1,I-1)-DQ(IA-J,J)*P(IA-J,I)
41.     110 IF(K2.LT.0)GO TO 120
42.     IF((K2.EQ.0).OR.(K2.EQ.1))GO TO 135
43.     K5=K2/2
44.     DO 3 I=1,K5
45.     IF(NDF(IA-J,J-I).NE.0)GO TO 130
46.     3 CONTINUE;GO TO 135
47.     130 R(J)=-DQ(IA-J,J)*Q(IA-J,0)
48.     DO 9 I=1,J-1
49.     9 R(I)=Q(IA-J+1,I-1)-DQ(IA-J,J)*Q(IA-J,I);GO TO 120
50.     C*****
51.     C      CAS OU LE POLYNOME EST SUR L'EST D'UN BLOC
52.     C*****
53.     102 CALL DETECT(IA-J,1,J,LA,L);NR1=NR(1);N1=N(L)
54.     350 FORMAT(3X,'LE POLYNOME ',I3,3X,I3,'NON O.R.')
55.     IS=IA-J-N1+NR1+1;IF(IA-J.GT.-NL)GO TO 103
56.     DO 6 I=0,IS
57.     6 W(IS-I)=DBR(N1-NR1-1+I,J);W(IS+1)=1
58.     DO 30 I=0,J-1
59.     T(I)=0;M2=MIN(IS+1,I);M1=MAX(0,I+1+IS-J);R(I)=0
60.     DO 30 L1=M1,M2
61.     30 T(I)=T(I)+P(IA-J,I-L1)*W(L1)
62.     DO 16 I=0,J-IS-2
63.     16 T(I)=T(I)+DCR(IA-J,J)*PR(IA-J,I);IF(K2.LT.0)GO TO 120
64.     C*****
65.     C      CALCUL DE L'ASSOCIE EN UTILISANT LA DEFINITION
66.     C*****
67.     135 DO 40 I5=0,J-1
68.     R(I5)=0
69.     DO 40 I6=0,I5
70.     40 R(I5)=R(I5)-T(I5-I6)*T1(-1-I6);GO TO 120
71.     103 A=DE(IA-J-1,J);IF(IA-J.LE.N1-NR1-1)A=-A
72.     DO 4 I=0,J-1
73.     R(I)=0
74.     4 T(I)=P(IA-J-1,I)+A*P(IA-J,I);IF(K2.LT.0)GO TO 120
75.     K5=K2/2;IF(K5.LE.0)GO TO 135
76.     DO 5 I=1,K5
77.     IF(NDF(IA-J,J-I).NE.0)GO TO 47
78.     5 CONTINUE;GO TO 135
79.     47 DO 42 I=0,J-1
80.     42 R(I)=Q(IA-J-1,I)+A*Q(IA-J,I)
81.     120 WRITE(MAT,300)KA,J
82.     270 DO 8 I=0,J
83.     PR(IA-J,I)=P(IA-J,I);P(IA-J,I)=T(I);Q(IA-J,I)=R(I)
84.     8 CONTINUE;IF(K2.LT.0)GO TO 301
85.     DO 46 I=0,J
86.     46 WRITE(MAT,320)R(I),I,T(I),I
87.     C      APPEL DE IMPADE
88.     CALL IMPADE(R,T,K2,J);GO TO 301

```



```

89. 101 P(IA-J,J)=0;PR(IA-J,J)=0;Q(IA-J,J)=0
90. 301 IF(IA-J.GT.-NL)J=J+1;GO TO 340
91. IA=IA+1;GO TO 100
92. C*****
93. C
94. C CALCUL DES POLYNOMES ORTHOGONAUX REGULIERS DE LA DEMI-TABLE INFERIEURE
95. C ET LEURS ASSOCIES
96. C
97. C*****
98. 360 IF(IA.GT.NL)RETURN
99. IF(IA.EQ.1)NL=NL
100. P(IA,0)=1;PR(IA,0)=0;Q(IA,0)=0;J=1
101. 370 KA=IA-J;IF(NDF(IA-J,J).EQ.0)WRITE(MAT,350)KA,J;GO TO 201
102. K2=IA+J;T(J)=1;R(J)=0;IF(NDF(IA-J+1,J-1).EQ.0)GO TO 202
103. T(0)=-DQ(IA-J,J)*P(IA-J,0);R(0)=0;IF(J.EQ.1)GO TO 240
104. DO 22 I=1,J-1
105. R(I)=0
106. 22 T(I)=P(IA-J+1,I-1)-DQ(IA-J,J)*P(IA-J,I)
107. 240 IF(KA.GT.0)GO TO 220
108. IF(IA-J.EQ.0)GO TO 260
109. IF((IA-J.EQ.-1).AND.(J.EQ.1))GO TO 260
110. IF(IA-J.LT.-2)K5=K2/2;GO TO 68
111. DO 55 I=1,J-1
112. IF(NDF(IA-J,J-1).NE.0)GO TO 54
113. 55 CONTINUE;GO TO 260
114. 68 DO 59 I=1,K5
115. IF(NDF(IA-J,J-1).NE.0)GO TO 54
116. 59 CONTINUE;GO TO 260
117. 54 R(0)=-DQ(IA-J,J)*Q(IA-J,0)
118. DO 23 I=1,J-1
119. 23 R(I)=Q(IA-J+1,I-1)-DQ(IA-J,J)*Q(IA-J,I);GO TO 220
120. C*****
121. C CALCUL DE L'ASSOCIE EN UTILISANT LA DEFINITION
122. C*****
123. 260 DO 43 I=0,J-1
124. R(I)=0
125. DO 43 L=0,J-1-I
126. 43 R(I)=R(I)+T(L)*T(L+1);GO TO 220
127. C*****
128. C CAS OU LE POLYNOME EST SUR L'EST D'UN BLOC
129. C*****
130. 202 CALL DETECT(IA-J,1,J,LA,L);NR1=NR(1);N1=N(L)
131. IS=IA-J-N1+NR1+1;IF(IA-J.GT.-NL)GO TO 203
132. DO 166 I=0,IS
133. 166 W(IS-I)=DBR(N1-NR1-1+I,J);W(IS+1)=1
134. DO 333 I=0,J-1
135. T(I)=0;M2=MIN(IS+1,I);M1=MAX(0,I+1+IS-J);R(I)=0
136. DO 333 L1=M1,M2
137. 333 T(I)=T(I)+P(IA-J,I-L1)*W(L1)
138. DO 116 I=0,J-IS-2
139. 116 T(I)=T(I)+DCR(IA-J,J)*PR(IA-J,I);GO TO 260
140. 203 A=DE(IA-J-1,J);IF(IA-J.LE.N1-NR1-1)A=-A
141. DO 24 I=0,J-1
142. R(I)=0
143. 24 T(I)=P(IA-J-1,I)+A*P(IA-J,I);IF(IA-J.GT.0)GO TO 220
144. IF(IA-J.LT.-2)K5=K2/2;GO TO 80
145. DO 82 I=1,J-1
146. IF(NDF(IA-J,J-1).NE.0)GO TO 38
147. 82 CONTINUE;GO TO 260
148. 80 DO 36 I=1,K5
149. IF(NDF(IA-J,J-1).NE.0)GO TO 38
150. 36 CONTINUE;GO TO 260
151. 38 DO 35 I=0,J-1
152. 35 R(I)=Q(IA-J-1,I)+A*Q(IA-J,I)
153. 220 WRITE(MAT,300)KA,J
154. DO 28 I=0,J
155. PR(IA-J,I)=P(IA-J,I);P(IA-J,I)=T(I);Q(IA-J,I)=R(I)
156. 28 CONTINUE;IF(IA-J.GT.0)GO TO 401
157. DO 66 I=0,J
158. 66 WRITE(MAT,320)R(I),I,T(I),I
159. C
160. C APPEL DE IMPADE
161. CALL IMPADE(R,T,K2,J);GO TO 401
162. 201 P(IA-J,J)=0;PR(IA-J,J)=0;Q(IA-J,J)=0
163. 401 IF(J.LT.NL)J=J+1;GO TO 370
164. IA=IA+1;NL=NL-1;GO TO 360
165. 300 FORMAT(1X,'(',I3,',',I3,')',2X,'ASSOCIE',8X,'POLYNOME O,R,')
166. 320 FORMAT(10X,D22.15,'X**',I3,8X,D22.15,'T**',I3)
167. END

```

```

1.      SUBROUTINE IMPADE(X,Y,KA,J)
2.      C*****
3.      C    CE SOUS PROGRAMME IMPRIME L'APPROXIMANT DE PADE EN DEUX POINTS A PARTIR
4.      C    DU PLOYNAME ORTHOGONAL -Y- ET SON ASSOCIE -X-
5.      C    KA EST LE NUMERO DE LA DIAGONALE DE -Y-
6.      C    J EST LE DEGRE DE -Y-
7.      C*****
8.      C*****CARTE POUVANT ETRE MODIFIEE PAR L'UTILISATEUR*****
9.      DIMENSIONX(0:J),Y(0:J)
10.     MAT=108
11.     C*****FIN*****
12.     WRITE(MAT,20)KA,J;WRITE(MAT,500);I1=J;I2=J-1
13.     DO 1 I=0,J-1
14.     WRITE(MAT,200)X(I),I,Y(I1,I);I2=I2-1;I1=I1-1
15.     1 CONTINUE;WRITE(MAT,110)Y(0),J;RETURN
16.     20 FORMAT(/,3X,'APPROXIMANT DE PADE EN DEUX POINTS(',I3,',',I3,')',/
17.     1)
18.     200 FORMAT(20X,D22,15,'X**',I3,45X,D22,15,'X**',I3)
19.     500 FORMAT(15X,'NUMERATEUR',70X,'DENOMINATEUR')
20.     110 FORMAT(93X,D22,15,'X**',I3)
21.     RETURN
22.     END

```



SIXIEME PARTIE

APPROXIMANTS DES SERIES DE FONCTIONS

On va proposer un ensemble de sous-programmes qui calcule les approxi-
mants des séries de fonctions. On va aussi aborder le calcul des racines des
polynômes puisque l'on a besoin des racines des polynômes orthogonaux dans
le calcul des approximants des séries de fonctions.

On commence par donner les définitions et les propriétés qui sont uti-
lisées dans la mise en oeuvre.

Dans la deuxième partie on propose une méthode qui calcule les racines
des polynômes.

Enfin on présente un ensemble des sous-programmes qui calcule les ap-
proximants des séries de fonctions.

I. DEFINITIONS ET RAPPELS

Définition 6.1. : On appelle série de fonctions une série de la forme

$$f(t) = \sum_{i=0}^{\infty} c_i g_i(t) \text{ où les } g_i(t) \text{ sont des fonctions de } t.$$

On peut aussi définir $f_n(t) = \sum_{j=n}^{\infty} c_j g_j(t)$, $\forall n \in \mathbb{Z}$, avec la convention

$$g_j(t) = 0 \text{ et } c_j = 0 \quad \forall j < 0$$

et les fonctionnelles linéaires $c^{(j)}$, pour $j \in \mathbb{Z}$, sur l'espace vectoriel
des polynômes réels par $c^{(j)}(x^i) = c_{j+i}$, $\forall i \in \mathbb{N}$.

Définition 6.2. : Une fonction $G_n(x, t)$ telle que $c^{(n)}(G_n(x, t)) = f_n(t)$ sera appelée fonction génératrice de $f_n(t)$.

Remarque 6.3. : Dans toute la suite on supposera connues les fonctions génératrices des $f_n(t)$, et l'on sait que l'on peut les écrire sous la forme

$$G_n(x, t) = \sum_{j=n}^{\infty} x^{j-n} g_j(t).$$

Définition 6.4. : Soient $p \geq 1$, $0 \leq k \leq p-1$, les $(x_i)_{0 \leq i \leq k}$ $k+1$ points distincts et les $(n_i)_{0 \leq i \leq k}$ tels que $\sum_{j=0}^k n_j = p$ avec $n_j \geq 1 \quad \forall j$.

La fonction $H(t) = \sum_{i=0}^k \sum_{j=0}^{n_i-1} \lambda_{i,j} G^{(j)}(x_i, t)$, où $G^{(j)}$ est la dérivée

$j^{\text{ième}}$ par rapport à x de la fonction génératrice de f , est appelée approximant de la série de fonctions f et est notée $[p-1/p]_f$ si

$$f(t) - H(t) = O(g_{2p}(t))$$

Définition 6.5. : Soient f une fonction analytique régulière dans un domaine connexe, les x_i pour $i \in \mathbb{N}$, $1 \leq i \leq m$, m points distincts intérieurs à ce domaine et, les n_i pour $i \in \mathbb{N}$, $1 \leq i \leq m$, m entiers supérieurs à 1 tels que

$$n = \sum_{i=1}^m n_i.$$

On appelle polynôme d'interpolation d'Hermite de f basé sur les (x_i, n_i) pour $i \in \mathbb{N}$, $1 \leq i \leq m$, le polynôme $L_n(x)$ tel que

$$L_n^{(k)}(x_j) = f^{(k)}(x_j) \quad \forall j \in \mathbb{N}, 1 \leq j \leq m \\ \text{et } k \in \mathbb{N}, 0 \leq k \leq n_j-1$$

Théorème 6.6. : Si $G_n(x, t)$ est la fonction génératrice de f_n alors

$$[p-1/p]_{f_n}(t) = c^{(n)}(L_{p,n}(x))$$

où $L_{p,n}(x)$ est le polynôme d'interpolation d'Hermite de $G_n(x, t)$ basé sur les $(m_{j,p}, q_{j,p})$ pour $j \in \mathbb{N}$, $0 \leq j \leq \ell_p$, les $m_{j,p}$ sont les racines distinctes du polynôme $P_p^{(n)}(x)$ orthogonal par rapport à $c^{(n)}$ et les $q_{j,p}$ sont les ordres de multiplicité des racines. C'est-à-dire $L_{p,n}$ est tel que

$$L_{p,n}^{(k)}(m_{j,p}) = G_n^{(k)}(m_{j,p}, t) \text{ pour } j \in \mathbb{N}, 0 \leq j \leq \ell_p \\ \text{et } k \in \mathbb{N}, 0 \leq k \leq q_{j,p} - 1$$

avec $G_n^{(k)}(m_{j,p}, t)$ la dérivée $k^{\text{ième}}$ de $G_n(x, t)$ par rapport à x au point $(m_{j,p}, t)$.

$$\text{On peut écrire } [p-1/p]_{f_n}(t) = \sum_{j=0}^{\ell_p} \sum_{s=0}^{q_{j,p}-1} K_{i_j+s,p} G_n^{(s)}(m_{j,p}, t) \quad (1)$$

où la dérivée $s^{\text{ième}}$ de G_n est prise par rapport à x et les i_j sont tels que

$$\begin{cases} i_0 = 0 \\ i_j = \sum_{s=0}^{j-1} q_{s,p} \text{ pour } j \in \mathbb{N}, 1 \leq j \leq \ell_p + 1. \end{cases}$$

Donnons les théorèmes 7.1 et 7.2 de [1].

Théorème 6.7. [1, page 557] : Les $K_{j,p}$ et $m_{j,p}$ de (1) satisfont le système (2) suivant :

$$(2) \quad \begin{pmatrix} 1 & 0 & 0 & \cdots & 1 & \cdots & 0 \\ m_{0,p} & 1 & 0 & \cdots & m_{1,p} & \cdots & 0 \\ m_{0,p}^2 & 2m_{0,p} & 2 & \cdots & m_{1,p}^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ m_{0,p}^{2p-1} & (2p-1)m_{0,p}^{2p-2} & (2p-2)(2p-1)m_{0,p}^{2p-3} & \cdots & m_{1,p}^{2p-1} & \cdots & (2p-1) \cdots (2p-q_{\ell_p,p}+1)m_{\ell_p,p}^{2p-q_{\ell_p,p}} \end{pmatrix} \begin{pmatrix} k_{0,p} \\ \vdots \\ k_{p-1,p} \end{pmatrix} = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{2p-1} \end{pmatrix}$$

Théorème 6.8. [1, page 564] : Si le polynôme P_p orthogonal par rapport à c existe, alors l'approximant $[p-1/p]_f$ existe et est unique.

Si le polynôme P_p est quasi-orthogonal, l'approximant $[p-1/p]_f$ n'existe pas.

Corollaire 6.9. [1, page 565] : Si le polynôme P_p est orthogonal singulier alors l'approximant $[p-1/p]_f$ est identique à l'approximant $[pr(p)-1/pr(p)]_f$.

On définit les approximants $[n+p-1/p]_f(t)$, $\forall n \in \mathbb{Z}$, $n \geq 1-p$ et $p \in \mathbb{N}$ par

$$\begin{aligned} [n+p-1/p]_f(t) &= \sum_{i=0}^{n-1} c_i g_i(t) + c^{(n)}_{(L_{p,n}(x))} \\ &= \sum_{i=0}^{n-1} c_i g_i(t) + [p-1/p]_{f_n}(t) \end{aligned}$$

Théorème 6.10. [1, page 558] :

$$f(t) - [n+p-1/p]_f(t) = O(g_{n+2p}(t)), \forall n \in \mathbb{Z}, n \geq 1-p$$

Définition 6.11. : On appellera table A la table à double entrée dans laquelle sont rangés les approximants $[p/q]_f$ de la manière suivante :

[0/0]	[0/1]	[0/2]	
[1/0]	[1/1]	[1/2]	
[2/0]	[2/1]	[2/2]	
.	.	.	
.	.	.	
.	.	.	
.	.	.	

Définition 6.12. : On appellera bloc A un bloc carré situé dans la table A qui correspond au bloc P et à ses côtés Nord, Nord-Ouest et Ouest.

Théorème 6.13. [1, page 575] : Dans un bloc A, au dessus de l'antidiagonale principale et sur cette dernière les approximants sont tous identiques.

II. CALCUL DES RACINES DES POLYNOMES

On peut calculer les racines d'un polynôme avec les méthodes de Bairstow et de Newton : la première donnant les valeurs à partir desquelles on génère, à l'aide de la deuxième, les suites qui convergent vers les racines du polynôme. Cette combinaison de ces deux méthodes donne des très bons résultats pour les racines simples, les racines multiples ne sont pas toujours calculées avec beaucoup de chiffres significatifs puisque la convergence des suites est souvent très lente.

Pour calculer les racines du polynôme $P(x)$ on génère les suites à l'aide d'une variante de la méthode de Newton avec la relation de récurrence

$$x_{j+1} = x_j - \Delta x_j \text{ où } \Delta x_j = \frac{h(x_j)}{h'(x_j)} \text{ avec } h(x) = \frac{P(x)}{P'(x)}.$$

On démontre que la méthode ci-dessus est quadratique.

On peut encore écrire Δx_j sous les formes

$$\Delta x_j = \frac{P'(x_j) P(x_j)}{(P'(x_j))^2 - P''(x_j) P(x_j)} \quad (1)$$

et

$$\Delta x_j = \frac{1}{\frac{P'(x_j)}{P(x_j)} - \frac{P''(x_j)}{P'(x_j)}} \quad (2)$$

L'écriture (2) est plus stable car si $P(x)$ a des racines multiples Δx_j est égal à l'inverse de la différence de deux grands nombres alors qu'avec (1) Δx_j est le quotient de deux nombres très petits.

Dans l'écriture (1) les produits $P'(x_j) P(x_j)$, $(P'(x_j))^2$ ou $P''(x_j) P(x_j)$ peuvent entraîner un dépassement de la capacité machine ($[10^{-75}, 10^{75}]$ pour l'IRIS 80) et donc ne pas permettre l'obtention d'un assez grand nombre de chiffres significatifs pour les racines multiples. On va utiliser (2) parce qu'elle est plus stable et elle peut, dans certains cas, donner des résultats très frappants (voir dernier exemple du tableau des essais numériques).

On propose une méthode qui calcule le polynôme $Q(x)$ de plus petit degré qui a toutes ses racines simples et les mêmes qu'un polynôme donné $P(x)$. On détermine facilement les racines de $P(x)$ à partir de $Q(x)$ à l'aide des méthodes de Bairstow et de Newton.

Le polynôme $Q(x)$ peut être obtenu à l'aide de la fraction continue associée au rapport $\frac{P(x)}{P'(x)}$. En effet le polynôme $Q(x)$ tel que

$$\frac{Q(x)}{R(x)} = a_0 x + \alpha_0 + \frac{1}{a_1 x + \alpha_1 + \frac{1}{a_2 x + \alpha_2 + \frac{1}{\dots}}} \quad (3)$$

le second membre de (3) étant la fraction continue associée au rapport $\frac{P(x)}{P'(x)}$,

a toutes ses racines simples et les mêmes que $P(x)$ puisque $\frac{P(x)}{P'(x)} = \frac{Q(x)}{R(x)}$.

Pour illustrer ceci, cherchons le polynôme $Q(x)$ qui a toutes ses racines simples et les mêmes que $P(x) = (x+1)^8 (x-1)$

$$\begin{aligned} &= x^9 + 7x^8 + 20x^7 + 28x^6 + 14x^5 - 14x^4 - 28x^3 \\ &\quad - 20x^2 - 7x - 1. \end{aligned}$$

$$\text{On a } \frac{P(x)}{P'(x)} = \frac{1}{9}x + \frac{7}{81} + \frac{-32/81}{9x-7} = \frac{x^2-1}{9x-7}$$

et donc $Q(x) = x^2 - 1$.

Le tableau suivant présente les résultats de quelques essais numériques.

Les racines sont données sous forme complexe : d'abord la partie réelle et ensuite la partie imaginaire.

Méthodes Polynôme	Bairstow et Newton appliquées directement au polynôme	Calcul de Q(x) et des racines de ce dernier par Bairstow et Newton
$(x+1)^2(x-1)(x^2-4)$	$-.2000000000000000+01$ $.0000000000000000+0$ $-.1000000005496210+01$ $.0000000000000000+0$ $-.2000000000000000+01$ $.0000000000000000+0$ $-.1000000001364710+01$ $.0000000000000000+0$ $-.1000000000000000+01$ $.0000000000000000+0$	Q(x) n'a pas pu être calculé
$(x+1)^2(x-1)^2(x+2)$	$-.2000000000000000+01$ $.0000000000000000+0$ $-.1000000000031080+01$ $.0000000000000000+0$ $-.1000000003353100+01$ $.0000000000000000+0$ $-.1000000001055770+01$ $.0000000000000000+0$ $-.1000000000211810+01$ $.0000000000000000+0$	$-.2000000000000000+01$ $.0000000000000000+0$ $-.1000000000000000+01$ $.0000000000000000+0$ $-.1000000000000000+01$ $.0000000000000000+0$
$(x+1)^3(x-1)^2$	$-.1000000132999080+01$ $.0000000000000000+0$ $-.10000001702297630+01$ $.0000000000000000+0$ $-.9999999999996050+00$ $.0000000000000000+0$ $-.10000001732679030+01$ $.0000000000000000+0$ $-.1000000000127460+01$ $.0000000000000000+0$	$-.1000000000000000+01$ $.0000000000000000+0$ $-.1000000000000000+01$ $.0000000000000000+0$
$(x+1)^4(x+2)$	$-.2000000000000000+01$ $.0000000000000000+0$ La deuxième suite générée avec la méthode de Newton ne converge pas.	$-.2000000000000000+01$ $.0000000000000000+0$ $-.1000000000000000+01$ $.0000000000000000+0$
$(x+1)^5$	$-.9999999999999040+00$ $.0000000000000000+0$ La deuxième suite générée avec la méthode de Newton ne converge pas.	$-.1000000000000000+01$ $.0000000000$ $.0000+0$

On arrête la méthode de Newton à 300 itérations si aucun des deux tests d'arrêt $\left| \frac{P(x)}{P'(x)} \right| \leq 10^{-15}$ et $\left| \frac{\Delta x}{x} \right| \leq 10^{-15}$ n'est satisfait.

On remarque que des résultats très excellents sont obtenus chaque fois que l'on a pu calculer le polynôme Q(x).



La précision des racines de $P(x)$ calculées à partir de $Q(x)$ dépend évidemment de la précision dans le calcul de ce dernier. D'une manière générale une assez bonne précision de $Q(x)$ s'avère difficile, sinon impossible quand son degré est assez grand car la fraction continue de (3) ou le rapport $\frac{Q(x)}{R(x)}$ entraînent un dépassement de la capacité machine ($[10^{-75}, 10^{75}]$ pour l'IRIS 80).

Dans le cas où les coefficients du polynôme $P(x)$ sont rationnels le calcul de $Q(x)$ devrait se faire sans aucune difficulté avec une arithmétique rationnelle, il suffit de prendre des dispositions pour que dans le calcul de la fraction continue de (3) et dans celui du rapport $\frac{Q(x)}{R(x)}$ les numérateurs et dénominateurs ne dépassent pas trop vite la capacité machine.

III. LA MISE EN OEUVRE

Elle se fait avec les sous-programmes QUDE, EST, DETECT, SERFON, GAUSS, MATRI, BAIRNE et RACINE. Dans le sous-programme SERFON les polynômes sont calculés suivant l'algorithme de calcul des polynômes orthogonaux réguliers de III et sont notés comme dans APADE de III.

Les sous-programmes EST, DETECT, QUDE et SERFON, et le programme principal s'échangent les valeurs des c_i , $q_k^{(n)}$, $e_k^{(n)}$, $B_k^{(n)}$ et $C_k^{(n)}$, et le tableau NDF par l'intermédiaire de l'instruction COMMON.

MODE D'EMPLOI DES SOUS-PROGRAMMES

* * * SERFON * * * *

Appel : CALL SERFON(NL)

Paramètre : Le tableau des coefficients c_i est indicé de 0 à NL.

NL est un paramètre d'entrée.

Sous-programmes utilisés : QUDE, BAIRNE, GAUSS, MATRI, EST, DETECT et RACINE.

Description : Ce sous-programme calcule les approximants $[p-1/p]$ des séries de fonctions f_n .

* * * GAUSS * * * *

Appel : CALL GAUSS(Q, T, N, X)

Paramètres : Q est une matrice carrée (N, N) à coefficients complexes indicés à partir de zéro.

T (resp. X) est un tableau indicé de 0 à N-1 et à coefficients réels (resp. complexes).

Q, T et N sont des paramètres d'entrée et X est un paramètre de sortie.

Description : Résolution du système $QX = T$ par la méthode de GAUSS.

* * * MATRI * * * *

Appel : CALL MATRI(X, IQ, N, IK, Q)

Paramètres : X (resp. IQ) est un tableau indicé de 0 à IK et à coefficients complexes (resp. entiers ≥ 1)

N est un entier tel que $\sum_{I=0}^{IK} IQ(I) = N$.

Q est une matrice carrée (N, N) à coefficients complexes indicés à partir de zéro.

X, IQ, N et IK sont des paramètres d'entrée et Q est un paramètre de sortie.

Description : Ce sous-programme calcule la matrice associée au système qui donne les coefficients l'approximant $[N-1/N]$ de la série de fonctions.

* * * BAIRNE * * * *

Appel : CALL BAIRNE(Y, N, X, IQ, IK)

Paramètres : Y est le tableau des coefficients du polynôme, il est indicé de 0 à N. Y(I) est le coefficient de x^I .

X (resp. IQ) est un tableau à coefficients complexes (resp. entiers ≥ 1) indicé de 0 à N.

IK est un entier ≥ 0 .

Y et N sont des paramètres d'entrée et, X, IQ et IK sont des paramètres de sortie.

Sous-programme utilisé : RACINE.

Description : Ce sous-programme calcule les racines du polynôme et leurs ordres, éventuellement en utilisant un polynôme de plus petit degré qui à toutes ses racines simples et les mêmes que le polynôme de départ.

IK+1 est le nombre de racines distinctes. Les racines et leurs ordres sont rangés dans les tableaux X et IQ tels que IQ(I) est l'ordre de X(I) pour $0 \leq I \leq IK$.

Ce sous-programme est conçu pour $N \leq 5$.

* * * RACINE * * * *

Appel : CALL RACINE(Y, N, X)

Paramètres : Y est le tableau des coefficients du polynôme, il est indicé de 0 à N. Y(I) est le coefficient de x^I .

X est un tableau à coefficients complexes, indicé de 0 à N.

Y et N sont des paramètres d'entrée et X est un paramètre de sortie.

Description : Calcul des racines du polynôme avec les méthodes de Bairstow et de Newton : la première donne les valeurs à partir desquelles on génère les suites à l'aide de la variante de Newton ((2), page 120).

Les autres sous-programmes sont dans II.

EXEMPLE D'APPEL DU SOUS-PROGRAMME SERFON

On donne un programme qui lit les valeurs de NL et des c_i et, fait appel au sous-programme SERFON.

```

1.  C*****
2.  C  PROGRAMME PRINCIPAL-CALCUL DES APPROXIMANTS D'UNE SERIE DE FONCTIONS
3.  C*****
4.  C*****CARTES SUSCEPTIBLES D'ETRE MODIFIEES PAR L'UTILISATEUR*****
5.  CCM40NDCR(-20;20,J:20),DC(-20;20),DBR(-20;20,0:20),DB(-20;20,0:20)
6.  1,DE(-20;20,0:20),WDF(-20;20,0:20)/S/N(20),K(20),NR(20)
7.  LGB=105;MAT=108
8.  C*****FIN*****
9.  READ(LGB,10)NL
10. READ(LGB,20)(DC(I),I=0,NL)
11. 20 FORMAT(10F15.8)
12. 10 FORMAT(I5)
13. CALL SERFON(NL);STOP
14. END

```

ESSAI NUMERIQUE

Pour NL = 10 et les coefficients c_i suivants :

```

C( 0)= .100000000000000000+01
C( 1)= .200000000000000000+01
C( 2)= .500000000000000000+01
C( 3)= .100000000000000000+02
C( 4)= .170000000000000000+02
C( 5)= .260000000000000000+02
C( 6)= .370000000000000000+02
C( 7)= .500000000000000000+02
C( 8)= .650000000000000000+02
C( 9)= .820000000000000000+02
C(10)= .101000000000000000+03

```


où les a_i sont des complexes avec la partie réelle sur la première colonne et la partie imaginaire sur la deuxième,

et $G(j_k, x_{I_k})$ est la dérivée j_k^e de $G(x, t)$ par rapport à x au point

(x_{i_k}, t) avec $G(x, t)$ la fonction génératrice de f_l ,

est égal à $\sum_{s=0}^{p-1} a_s G(j_s, x_{I_k})$.

$[2/3]_{f_D} =$

.1000000000000000+01	.0000000000000000+00	*G(0, .1000000000000000+01	.0000000000000000+00)
.1000000000000000+01	.0000000000000000+00	*G(1, .1000000000000000+01	.0000000000000000+00)
.1000000000000000+01	.0000000000000000+00	*G(2, .1000000000000000+01	.0000000000000000+00)

$[2/3]_{f_1} =$

.2000000000000000+01	.0000000000000000+00	*G(0, .1000000000000000+01	.0000000000000000+00)
.3000000000000000+01	.0000000000000000+00	*G(1, .1000000000000000+01	.0000000000000000+00)
.1000000000000000+01	.0000000000000000+00	*G(2, .1000000000000000+01	.0000000000000000+00)

$[2/3]_{f_2} =$

.5000000000000000+01	.0000000000000000+00	*G(0, .1000000000000000+01	.0000000000000000+00)
.4999999999999999+01	.0000000000000000+00	*G(1, .1000000000000000+01	.0000000000000000+00)
.9999999999999999+00	.0000000000000000+00	*G(2, .1000000000000000+01	.0000000000000000+00)

$[2/3]_{f_3} =$

.1000000000000000+02	.0000000000000000+00	*G(0, .9999999999999999+00	.0000000000000000+00)
.7000000000000000+01	.0000000000000000+00	*G(1, .9999999999999999+00	.0000000000000000+00)
.1000000000000000+01	.0000000000000000+00	*G(2, .9999999999999999+00	.0000000000000000+00)



$$[2/3]_{f_4} =$$

.1700000000000000+02	.0000000000000000+00	*G(0, .1000000000000080+01	.0000000000000000+00)
.8999999999999999+01	.0000000000000000+00	*G(1, .1000000000000080+01	.0000000000000000+00)
.9999999999999999+00	.0000000000000000+00	*G(2, .1000000000000080+01	.0000000000000000+00)

$$[2/3]_{f_5} =$$

.2600000000000000+02	.0000000000000000+00	*G(0, .9999999999998740+00	.0000000000000000+00)
.1100000000000000+02	.0000000000000000+00	*G(1, .9999999999998740+00	.0000000000000000+00)
.1000000000000000+01	.0000000000000000+00	*G(2, .9999999999998740+00	.0000000000000000+00)

On vérifie sans difficulté que les approximants $[i+2/3]_f$ pour $i = 0, 1, \dots, 5$ sont identiques.

CONCLUSION :

On a bien identifié des approximants $[p/q]_f$ à l'Ouest, au Nord et dans le coin Nord-Ouest d'un bloc P.


```

1.      SUBROUTINE SERFON(NL)
2.      C*****
3.      C      SOUS PROGRAMME-CALCUL DES APPROXIMANTS D'UNE SERIE DE FONCTIONS
4.      C*****
5.      C*****CARTES SUSCEPTIBLES D'ETRE MODIFIEES PAR L'UTILISATEUR*****
6.      COMMON/DCR(-20:20,0:20),CC(-20:20),DCR(-20:20,0:20),DD(-20:20,0:20),
7.      TDE(-20:20,0:20),NDF(-20:20,0:20)/S/N(20),K(20),NR(20)
8.      DIMENSIONP(0:20,0:10),PR(0:20,0:10),T(0:20),W(0:20),IQ(0:20)
9.      DIMENSIONY(0:20)
10.     COMPLEXX(0:10),AM1(0:10,0:10),LAND(0:10)
11.     MAT=108
12.     C*****
13.     IF(NL.LE.0)WRITE(MAT,600);STOP
14.     600 FORMAT(2X,'ERREUR CAR NL DOIT ETRE SUP A ZERO')
15.     C      APPEL A QUDE
16.     CALL QUDE(NL,0,0,LA);IA=0;ML=(NL+1)/2
17.     C*****
18.     C
19.     C      CALCUL DES POLYNOMES ORTHOGONAUX REGULIERS DE LA DEMI-TABLE SUPERIEURE
20.     C      ET DES APPROXIMANTS ASSOCIES A CES POLYNOMES
21.     C*****
22.     P(IA,0)=1;PR(IA,0)=0;IA=1
23.     100 IF(IA.GT.ML)IA=IA-1;GO TO 390
24.     P(IA,0)=1;PR(IA,0)=0;J=1
25.     340 KA=1A-J;IF(NDF(IA-J,J).EQ.0)K1=J-1;WRITE(MAT,350)K1,J,KA;GO TO 101
26.     T(J)=1;IF(NDF(IA-J+1,J-1).EQ.0)GO TO 102
27.     T(0)=-DD(IA-J,J)*P(IA-J,0);IF(J.EQ.1)GO TO 120
28.     DO 2 I=1,J-1
29.     2 T(I)=P(IA-J+1,I-1)-DD(IA-J,J)*P(IA-J,I);GO TO 120
30.     102 CALL DETECT(IA-J,1,J,LA,L);NR1=NR(1);N1=N(L)
31.     IS=IA-J-N1+NR1+1;IF(IA-J.GT.0)GO TO 103
32.     DO 6 I=0,IS
33.     6 W(IS-I)=DCR(N1-NR1-1+I,J);W(IS+1)=1
34.     DO 30 I=0,J-1
35.     T(I)=0;M2=MAX(0,I+IS+1-J);M3=MIN(I,IS+1)
36.     DO 30 L=M2,M3
37.     30 T(I)=T(I)+P(IA-J,I-L)*W(L)
38.     DO 16 I=0,J-IS-2
39.     16 T(I)=T(I)+DCR(IA-J,J)*PR(IA-J,I);GO TO 120
40.     103 A=DE(IA-J-1,J);IF(IA-J.LE.N1-NR1-1)A=-A
41.     DO 4 I=0,J-1
42.     4 T(I)=P(IA-J-1,I)+A*P(IA-J,I)
43.     120 K1=J-1;WRITE(MAT,300)K1,J,KA
44.     DO 8 I=0,J
45.     PR(IA-J,I)=P(IA-J,I);P(IA-J,I)=T(I)
46.     8 CONTINUE
47.     350 FORMAT(//,3X,'APPROXIMANT(',I2,'/',I2,')F',I2,'EST DANS UN BLOC')
48.     300 FORMAT(//,3X,'APPROXIMANT(',I2,'/',I2,')F',I2,/)
49.     C      CALCUL DES RACINES DES POLYNOMES
50.     CALL BAIRNE(T,J,X,I0,I2)
51.     C      DETERMINATION DES COEFFICIENTS DE LA MATRICE
52.     CALL MATN(X,I0,J,I2,AM1)
53.     DO 46 I=0,J-1
54.     46 Y(I)=C(IA-J+I)
55.     C      RESOLUTION DU SYSTEME D'EQUATIONS
56.     CALL GAUSS(AM1,Y,J,LAND);IK=0
57.     DO 50 I=0,I2
58.     DO 52 L=0,I2(I)-1
59.     52 WRITE(MAT,1002)LAND(IK+L),L,X(I)
60.     50 IK=IK+I2(I);GO TO 601
61.     101 P(IA-J,J)=0;PR(IA-J,J)=0
62.     801 IF(J.LT.1A)J=J+1;GO TO 340
63.     1A=1A+1;GO TO 100
64.     1002 FORMAT(10X,2(D22,15),5X,'+G(',I3,',',2(D22,15),')')
65.     C*****
66.     C
67.     C      CALCUL DES POLYNOMES ORTHOGONAUX REGULIERS DE LA DEMI-TABLE INFERIEURE
68.     C      ET DES APPROXIMANTS ASSOCIES A CES POLYNOMES
69.     C*****
70.     390 IF(MOD(NL,2).NE.0)ML=ML-1
71.     360 IA=IA+1;IF(IA.GT.NL)RETURN
72.     P(IA,0)=1;PR(IA,0)=0;J=1
73.     370 KA=1A-J;IF(NDF(IA-J,J).EQ.0)K1=J-1;WRITE(MAT,350)K1,J,KA;GO TO 201
74.     T(J)=1;IF(NDF(IA-J+1,J-1).EQ.0)GO TO 202
75.     T(0)=-DD(IA-J,J)*P(IA-J,0);IF(J.EQ.1)GO TO 220
76.     DO 22 I=1,J-1
77.     22 T(I)=P(IA-J+1,I-1)-DD(IA-J,J)*P(IA-J,I);GO TO 220
78.     202 CALL DETECT(IA-J,1,J,LA,L);NR1=NR(1);N1=N(L)
79.     A=DE(IA-J-1,J);IF(IA-J.LE.N1-NR1-1)A=-A
80.     DO 24 I=0,J-1
81.     24 T(I)=P(IA-J-1,I)+A*P(IA-J,I)
82.     220 K1=J-1;WRITE(MAT,300)K1,J,KA
83.     DO 28 I=0,J
84.     PR(IA-J,I)=P(IA-J,I);P(IA-J,I)=T(I)
85.     28 CONTINUE
86.     C      CALCUL DES RACINES DES POLYNOMES
87.     CALL BAIRNE(T,J,X,I0,I3)
88.     C      DETERMINATION DES COEFFICIENTS DE LA MATRICE

```




```

89.      CALL MATRI(X,IC,J,I3,AM1)
90.      DO 58 I=0,J-1
91.      50 Y(I)=C(IA-J+I)
92.      C      RESOLUTION DU SYSTEME D'EQUATIONS
93.      CALL GAUSS(AM1,Y,J,LAND);IK=0
94.      DO 60 I=0,I3
95.      DO 62 L=0,IQ(I)-1
96.      62 WRITE(MAT,1002)LAND(IK+L),L,X(I)
97.      60 IK=IK+IQ(I);GO TO 901
98.      201 P(IA-J,J)=C;PR(IA-J,J)=C
99.      901 IF(J.LT.ML)J=J+1;GO TO 370
100.      ML=ML-1;GO TO 360
101.      END

```

```

1.      SUBROUTINE MATRI(X,IQ,N,IK,Q)
2.      C*****
3.      C      CALCUL DES ELEMENTS DE LA MATRICE DU ASSOCIEE AU SYSTEME DONNANT
4.      C      LES COEFFICIENTS DES APPROXIMANTS DES SERIES DE FONCTIONS
5.      C*****
6.      C*****CARTE SUSCEPTIBLE D'ETRE MODIFIEE PAR L'UTILISATEUR*****S3
7.      COMPLEXX(0:4),Q(0:10,0:N)
8.      C*****FIN*****
9.      DIMENSIONIQ(0:N)
10.     IF(N.EQ.1)Q(0,0)=1;RETURN
11.     N=0
12.     DO 1 I=0,IK
13.     DO 2 J=1,N-1
14.     2 Q(J,M)=X(I)+J;Q(0,M)=1;IF(IQ(I).LT.2)GO TO 1
15.     DO 3 J=1,IQ(I)-1
16.     DO 4 L=0,N-1
17.     Q(L,M+J)=0;IF(L.GT.J-1)Q(L,M+J)=L*Q(L-1,M+J-1)
18.     4 CONTINUE
19.     3 CONTINUE
20.     1 M=M+IQ(I);RETURN
21.     END

```

```

1.      SUBROUTINE RACINE(Y,N,X)
2.      C*****
3.      C      CALCUL DES RACINES D UN POLYNOME PAR LES METHODES DE BAIRSTOW
4.      C      ET DE NEWTON
5.      C*****
6.      C***** CARTES POUVANT ETRE MODIFIEES PAR L'UTILISATEUR*****
7.      DIMENSIONY(0:N),B(0:10),T(0:10)
8.      COMPLEXX(0:N),DR,FO,Z,DW
9.      MAT=108
10.     EPS=1.D-15
11.     C***** FIN *****
12.     EP=1.D-2;IF(N.GT.0)GO TO 45
13.     505 FORMAT(3X,'LE DEGRE DU POLYNOME EST NEGATIF ERREUR')
14.     IF(N.LT.0)WRITE(MAT,1000);STOP
15.     IF(ABS(Y(0)).LT.1.D-12)WRITE(MAT,1000);STOP
16.     WRITE(MAT,800);STOP
17.     1000 FORMAT(3X,'LE POLYNOME A UNE INFINITE DE RACINES')
18.     C*****
19.     C      METHODE DE BAIRSTOW
20.     C*****
21.     45 DO 1 I=N,0,-1
22.     IF(ABS(Y(I)).GT.1.D-12)M=I;GO TO 51
23.     1 CONTINUE;WRITE(MAT,500);STOP
24.     500 FORMAT(3X,'TOUS LES COEFFICIENTS SONT NULS')
25.     51 DO 3 I=0,M
26.     X(M-I-1)=0;IF(ABS(Y(I)).GT.0)M2=1;M=M-M2;GO TO 50
27.     3 CONTINUE
28.     50 IF(M.EQ.0)M1=0;GO TO 60
29.     DO 2 J=0,M
30.     T(J)=Y(M2+J);I2=1;IV=1;M1=M;IF(M.GT.2)GO TO 55
31.     53 IF(M.EQ.1)X(0)=-T(0)/T(1);GO TO 60
32.     A=T(1)*T(1)-4*T(0)*T(2);X(1)=(-T(1)-CSQRT(A))/(2*T(2))
33.     X(0)=(-T(1)+CSQRT(A))/(2*T(2));GO TO 60
34.     55 S=I2;P=IV;MAX=1;B(M)=T(M);LAT=0
35.     58 B(M-1)=T(M-1)+S*T(M);IF(MAX.GT.300)GO TO 68
36.     600 FORMAT(3X,'NON CONVERGENCE DANS BAIRSTOW')
37.     DO 5 I=M-2,0,-1
38.     B(I)=T(I)+S*B(I+1)-P*B(I+2)
39.     C3=B(M);C2=B(M-1)+S*C3;IF(M-2,EQ.1) GO TO 50
40.     DO 6 I=M-2,2,-1
41.     C1=B(I)+S*C2-P*C3;C3=C2;C2=C1
42.     6 CONTINUE
43.     56 C1=C2+S-P*C3
44.     D5=3(0)+C3-B(1)+C2;DP=B(0)+C2-B(1)+C1;DT=C2+C2-C1*C3
45.     D5=D5/DT;DP=DP/DT;S=S+D5;P=P+DP;DT=(ABS(D5)+ABS(DP))/(ABS(S)+ABS(P
46.     1))
47.     IF((ABS(D5).LT.EPS).AND.(ABS(DP).LT.EPS))GO TO 67
48.     IF(ABS(DT).GT.EP)LAT=0;MAX=MAX+1;GO TO 58
49.     IF(LAT.LT.5)LAT=LAT+1;GO TO 58
50.     A=S+2-4*P;X(M-1)=(S+CSQRT(A))/2
51.     X(M-2)=(S-CSQRT(A))/2;M=M-2
52.     DO 15 I=0,M
53.     15 T(I)=B(I+2);IF(M.LE.2)GO TO 53
54.     I2=1;IV=1;GO TO 55
55.     60 IF((I2.EQ.0).AND.(IV.EQ.0))WRITE(MAT,600);STOP
56.     I2=0;IV=0;MAX=0;GO TO 55
57.     61 IF((I2.EQ.0).AND.(IV.EQ.1))WRITE(MAT,600);STOP
58.     I2=0;IV=1;MAX=0;GO TO 55
59.     C*****
60.     C      METHODE DE NEWTON
61.     C*****
62.     60 DO 20 I=0,N-1
63.     MAX=0;IF(N1.LE.2)GO TO 20
64.     24 FO=Y(0)+Y(1)*X(I)+Y(2)*X(I)**2;DR=Y(1)+Y(2)*X(I);DL=2*Y(2)
65.     DO 21 J=J,N
66.     FO=FO+Y(J)*X(I)**J;DR=DR+J*Y(J)*X(I)**(J-1)
67.     21 DJ=DW+J*(J-1)*Y(J)*X(I)**(J-2)
68.     Z=FO/DR;IF(CABS(Z).LT.1.D-15) GO TO 20
69.     Z=1./((DR/FO)-(DW/DR));X(I)=X(I)-Z;Z=Z/X(I)
70.     IF(CABS(Z).LT.1.D-15)GO TO 20
71.     MAX=MAX+1;IF(MAX.GT.300)WRITE(MAT,550);STOP
72.     GO TO 22
73.     20 WRITE(MAT,200)I,X(I)
74.     550 FORMAT(4X,'IL N Y A PAS CONVERGENCE DE LA METHODE DE NEWTON')
75.     800 FORMAT(3X,'LE POLYNOME NE PEUT AVOIR DE RACINE')
76.     200 FORMAT(3X,'X('',I2,'')=',2(D22,15))
77.     RETURN
78.     END

```



```

1.      SUBROUTINE BAIRNE(Y,N,X,IQ,IK)
2.      C*****
3.      C      CALCUL DES RACINES D'UN POLYNOME
4.      C*****
5.      COMPLEXX(0:N),DR,FO
6.      C*****
7.      DIMENSIONY(0:N),B(0:10),T(0:10),IQ(0:N),JQ(0:10),D(0:10,0:10)
8.      DIMENSIONC(0:10)
9.      MAT=108
10.     C*****
11.     IK=-1;IF(N.LT.0)WRITE(MAT,800);STOP
12.     800 FORMAT(1X,'ERREUR LE DEGRE EST NEGATIF')
13.     IF(ABS(Y(N)).LT.1.D-14)WRITE(MAT,601);STOP
14.     601 FORMAT(3X,'LE POLYNOME N EST PAS DE DEGRE ',I2)
15.     DO 3 I=0,N
16.     IF(ABS(Y(I)).GT.1.D-14)I2=I;N2=I2;GO TO 4
17.     3 CONTINUE
18.     4 IF(I2.EQ.N)X(0)=0;IK=0;IQ(0)=N;GO TO 45
19.     IK=0;IF(I2.EQ.0)IK=-1
20.     I1=N-I2;N1=I1;IF(I1.GT.2)GO TO 6
21.     IF(IK.EQ.0)X(1)=0;IQ(1)=I2
22.     IF(I1.EQ.1)X(0)=-Y(I2)/Y(I2+1);IK=IK+1;IQ(0)=1;GO TO 45
23.     A=Y(I2+1)**2-4*Y(I2+2)*Y(I2);IF(ABS(A).GT.1.D-30)GO TO 7
24.     X(0)=-Y(I2+1)/(2*Y(I2+2));IQ(0)=2;IK=IK+1;GO TO 45
25.     7 IF(IK.EQ.0)X(2)=0;IQ(2)=I2
26.     IK=IK+2;X(1)=(-Y(I2+1)+CSQRT(A))/(2*Y(I2+2));IQ(1)=1;IQ(0)=1
27.     X(0)=(-Y(I2+1)-CSQRT(A))/(2*Y(I2+2));GO TO 45
28.     6 DO 5 I=1,I1
29.     B(I)=Y(I2+I)
30.     T(I-1)=I*Y(I2+I);B(0)=Y(I2);I2=I1-1;K=0
31.     C*****
32.     C      CALCUL DU POLYNOME DE PLUS PETIT DEGRE QUI A LES MEMES RACINES
33.     C      QUE LE POLYNOME DONT ON CALCULE LES RACINES
34.     C*****
35.     66 JQ(K)=I1-I2;C(K)=T(I2);D(K,JQ(K))=B(I1)
36.     DO 8 I=1,JQ(K)
37.     A=B(I1-I)*T(I2)**I
38.     DO 9 J=0,I-1
39.     AR=1;IF(I-J-1.NE.0)AR=T(I2)**(I-J-1)
40.     A=A-T(I1-JQ(K)-I+J)*D(K,JQ(K)-J)*AR
41.     8 D(K,JQ(K)-I)=A
42.     DO 10 J=0,I2-1
43.     A=B(J)*T(I2)**(JQ(K)+1);M2=MIN(JQ(K),J)
44.     DO 11 I=0,M2
45.     AR=1;IF(I.NE.0)AR=T(I2)**I
46.     11 A=A-D(K,I)*T(J-I)*AR
47.     10 B(J)=A
48.     DO 12 J=I2-1,0,-1
49.     IF(ABS(B(J)).GT.1.D-12)I1=J;GO TO 13
50.     12 CONTINUE;IF(K.NE.0)GO TO 17
51.     IF(IK.EQ.0)X(1)=0;IQ(1)=N-I1
52.     X(0)=-D(0,0)/(D(0,1)*C(0));IK=IK+1;IQ(0)=I1;GO TO 45
53.     13 DO 14 I=0,I2
54.     14 T(I)=T(I)*(T(I2)**(JQ(K)+1));IF(I1.EQ.0)A=B(0);GO TO 19
55.     DO 16 I=0,I2
56.     A=T(I);IF(I.LE.I1)T(I)=B(I)
57.     16 B(I)=A;I3=I1;I1=I2;I2=I3;K=K+1;JQ(K)=I1-I2;GO TO 66
58.     17 T(0)=1;I4=0;A=0;GO TO 20
59.     19 IF(K.LE.N1-2)GO TO 20
60.     C      CAS OU LE POLYNOME N'A PU ETRE CALCULE OU ENCORE A ETE CALCULE
61.     C      ET EST DE MEME DE DEGRE QUE LE POLYNOME DONT ON CALCULE LES RACINES
62.     DO 21 I=0,N1
63.     I2(I)=1;B(I)=Y(N2+I)
64.     21 CONTINUE;CALL RACINE(B,N1,X);IK=IK+N1;IF(IA.NE.0)GO TO 30
65.     IF(IK.EQ.N-1)GO TO 45
66.     GO TO 33
67.     20 DO 22 I=0,N1
68.     B(I)=0;IF(I.LE.I2)B(I)=T(I)
69.     T(I)=0;IF(I.EQ.0)T(I)=A
70.     22 CONTINUE;I0=I2;IN=0;IA=0
71.     DO 18 L=K,0,-1
72.     IN=I0+JQ(L)
73.     DO 23 I=JQ(L),1,-1
74.     23 D(L,I)=D(L,I)-C(L)**I;C(L)=C(L)**(JQ(L)+1)
75.     DO 25 I=0,IN
76.     T(I)=T(I)*C(L);M2=MAX(0,I-JQ(L));M3=MIN(I0,I)
77.     DO 25 J=M2,M3
78.     T(I)=T(I)+C(L,I-J)*B(J);IF(ABS(T(I)).GT.1.D+00)N2=0;IA=2;GO TO 28
79.     25 CONTINUE;IF(L.EQ.0)GO TO 30
80.     DO 26 I=0,I0
81.     B(I)=B(I)+C(L);IF(ABS(B(I)).GT.1.D+00)N2=0;IA=2;GO TO 28
82.     26 CONTINUE
83.     DO 27 I=0,IN
84.     A=T(I);T(I)=B(I);B(I)=A
85.     27 CONTINUE;I0=IN
86.     18 CONTINUE
87.     30 IF(IN.EQ.N)N2=0;GO TO 28

```

```

88. C CAS OU LE POLYNOME DE PLUS PETIT DE DEGRE EST DE DEGRE INFERIEUR
89. C A CELUI DU POLYNOME DONT ON CALCULE LES RACINES
90. IF(IN.GT.2)CALL RACINE(T,IN,X);IK=IK+IN;GO TO 33
91. A=T(1)+2-4*T(2)+T(0);X(0)=(-T(1)+CSQRT(A))/(2+T(2))
92. X(1)=(-T(1)-CSQRT(A))/(2+T(2))
93. IK=IK+2;IF(IK.EQ.2)X(2)=0;IQ(2)=N-1
94. 33 DO 34 I=0,IK
95. 34 WRITE(MAT,900)I,X(I)
96. 900 FORMAT(2X,'X(',I2,')=' ,2(D22.15))
97. DO 35 I=0,IK
98. IF(CABS(X(I)).LT.1,D-16)GO TO 35
99. DR=Y(1)+2*Y(2)+X(I);FO=2*Y(2)
100. DO 36 J=3,N
101. DR=DR+J*Y(J)*X(I)**(J-1);FO=FO+J*(J-1)*Y(J)*X(I)**(J-2)
102. 36 CONTINUE;IQ(I)=1
103. IF(CABS(FO).LT.1,D-15)IQ(I)=3;GO TO 35
104. IF(CABS(DR).LT.1,D-14)IQ(I)=2
105. 35 CONTINUE
106. IF(IK.GT.1)GO TO 45
107. IF(IQ(0)+IQ(1).GE.N)GO TO 45
108. IF(IQ(0).GT.IQ(1))IQ(0)=IQ(0)+1;GO TO 45
109. IQ(1)=IQ(1)+1;GO TO 45
110. 50 WRITE(MAT,900);JK=IK;I1=0
111. 900 FORMAT(3X,'LES RACINES D ORDRE MULTIPLE SONT TROUVEES A PARTIR DU
112. 18-EME DECIMAL')
113. DO 52 I=0,JK-1
114. IF(IQ(I).EQ.0)GO TO 52
115. DO 53 J=I+1,JK
116. IF(IQ(J).EQ.0)GO TO 53
117. IF(CABS(X(I)-X(J)).LT.1,D-7)IQ(J)=0;IQ(I)=IQ(I)+1;IK=IK-1
118. 53 CONTINUE;X(I1)=X(I);I1=I1+1
119. 52 CONTINUE;IF(IQ(JK).NE.0)X(I1)=X(JK);IQ(I1)=1
120. DO 51 I=0,IK
121. IF(IQ(I).LE.1)GO TO 51
122. A=REAL(X(I));G=AIMAG(X(I));A1=A*MOD(A,1,D-7);G1=G*MOD(G,1,D-7)
123. IF(ABS(A1).GT.1,D-15)A=A-AMOD(A,1,D-7)
124. IF(ABS(G1).GT.1,D-15)G=G-AMOD(G,1,D-7)
125. X(I)=CMPLX(A,G)
126. 51 CONTINUE
127. 45 DO 46 I=0,IK
128. 46 WRITE(MAT,400)I,X(I),IQ(I);RETURN
129. 400 FORMAT(3X,'X(',I5,')=' ,2(D22.15),20X,'MULTI=' ,I2)
130. END

```

```

1. SUBROUTINE GAUSS(W,T,N,X)
2. C*****
3. C RESOLUTION D'UN SYSTEME D'EQUATIONS PAR LA METHODE DE GAUSS
4. C*****
5. C***** CARTES POUVANT ETRE MODIFIEES PAR L'UTILISATEUR*****
6. COMPLEXD(0:10,0:N);X(0:N);B(0:10)
7. MAT=108
8. EPS=1.D-15
9. C*****
10. DIMENSIONT(0:N)
11. COMPLEXD,DA
12. DO 33 I=0,N-1
13. 33 B(I)=CMPLX(T(I),0);L=N-1
14. IF(L)91,92,90
15. 91 WRITE(MAT,95);STOP
16. 95 FORMAT(3X,'ERREUR SUR L ORDRE DE LA MATRICE')
17. 90 DO 4 L=N-1,1,-1
18. IM=L;DC=0
19. DO 2 J=L,0,-1
20. IF(CABS(Q(J,L)).LT.DC)GO TO 2
21. DC=CABS(Q(J,L));IM=J
22. 2 CONTINUE;IF(DC.LT.EPS)WRITE(MAT,200);STOP
23. 200 FORMAT(3X,'MATRICE ASSOCIEE SINGULIERE')
24. IF(IM.EQ.L)GO TO 110
25. DO 3 J=0,L
26. D=Q(IM,J);Q(IM,J)=Q(L,J);Q(L,J)=D
27. 3 CONTINUE;DA=B(L);B(L)=B(IM);B(IM)=DA
28. 110 DO 4 I=0,L-1
29. B(I)=B(I)-(Q(I,L)*B(L)/Q(L,L))
30. DO 4 J=0,L-1
31. 4 Q(I,J)=Q(I,J)-(Q(I,L)*Q(L,J)/Q(L,L))
32. 92 IF(CABS(Q(0,0)).LT.EPS)WRITE(MAT,200);STOP
33. X(0)=B(0)/Q(0,0);IF(N.EQ.1)RETURN
34. DO 5 I=1,N-1
35. D=B(I)
36. DO 6 J=0,I-1
37. D=D-Q(I,J)*X(J);X(I)=D/Q(I,I)
38. 5 CONTINUE;RETURN
39. END

```



SEPTIEME PARTIE

ALGORITHME qd
I. DEFINITION ET RELATION AVEC L'ALGORITHME qd

Soit la série formelle $f(t) = \sum_{j=0}^{\infty} c_j t^j$. On peut définir les fonctionnelles linéaires $\gamma^{(m)}$ telles que

$$\gamma^{(m)}(x^j) = c_{m-j} \quad \text{pour } j \in \mathbb{N}, m-j \geq 0$$

On abrègera orthoronal régulier en O.R.

Si on note $w_k^{(m)}(x)$ les polynômes orthogonaux par rapport à la fonctionnelle $\gamma^{(m)}$, on aura pour $m-2k+2 > 0$

$$w_k^{(m)} \text{ est O.R. si } \begin{bmatrix} c_m & c_{m-k+1} \\ c_{m-1} & \vdots \\ \vdots & \vdots \\ c_{m-k+1} & c_{m-2k+2} \end{bmatrix} = H_k^{(m-2k+2)} \neq 0$$

$$w_k^{(m)} \text{ non O.R. sinon}$$

Dans toute la suite on supposera les polynômes orthogonaux $w_k^{(m)}(x)$ unitaires et donc si $w_k^{(m)}$ est O.R. on peut écrire

$$w_k^{(m)}(x) = \frac{1}{H_k^{(m-2k+2)}} \begin{bmatrix} c_m & c_{m-k} \\ c_{m-1} & c_{m-k-1} \\ \vdots & \vdots \\ c_{m-k+1} & c_{m-2k+1} \\ 1 & x^k \end{bmatrix}$$

Définition 7.1. : On appelle table W un tableau à double entrée dans lequel sont rangées les familles adjacentes des polynômes orthogonaux par rapport aux $\gamma^{(m)}$, $m \in \mathbb{N}^*$, de la manière suivante :

$$\begin{array}{cccc}
 w_1^{(1)} & & & \\
 w_1^{(2)} & w_2^{(3)} & & \\
 w_1^{(3)} & w_2^{(4)} & w_3^{(5)} & \\
 w_1^{(4)} & w_2^{(5)} & w_3^{(6)} & w_4^{(7)} \\
 \vdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \vdots
 \end{array}$$

où l'indice supérieur correspond à l'indice de l'antidiagonale et l'autre à celui de la colonne.

On sait d'après le théorème 1.5 de [1, page 54] qu'il existe une relation de récurrence, à trois termes entre trois polynômes consécutifs $w_k^{(m)}(x)$, dont les coefficients sont déterminés par les relations présentées dans ce théorème en remplaçant la fonctionnelle $c^{(n)}$ par la fonctionnelle $\gamma^{(m)}$.

Si on note $\tilde{\text{pr}}(i, m)$ le degré du polynôme O.R. prédécesseur de $W_i^{(m)}$ sur l'antidiagonale m , on peut écrire

$$W_i^{(m)}(x) = \underbrace{\alpha_i^{(m)}}_{i-\text{pr}(i,m)-1}(x) + \underbrace{\beta_i^{(m)}}_{\text{pr}(i,m)} W_{\text{pr}(i,m)}^{(m)}(x) + \underbrace{\gamma_i^{(m)}}_{\text{pr}(\text{pr}(i,m),m)} W_{\text{pr}(\text{pr}(i,m),m)}^{(m)}(x)$$

où $\tilde{\alpha}^{(m)}_{i-\text{pr}(i,m)-1}(x)$ est un polynôme unitaire de degré $i-\text{pr}(i,m)-1$.

On démarre la relation ci-dessus avec $w_{-1}^{(m)}(x) = 0$ et $w_0^{(m)}(x) = 1$.

Dans le cas où la table W est normale, on peut définir un algorithme qui ressemble à l'algorithme $\tilde{q}d$ en posant

$$B_{k+1}^{(m)} = -q_{k+1}^{(m)} - e_k^{(m)} \quad \text{et} \quad C_{k+1}^{(m)} = -e_k^{(m)} q_k^{(m)},$$

on démontre que les polynômes orthogonaux $W_k^{(m)}(x)$ vérifient les relations

$$W_i^{(m-1)}(x) = W_i^{(m)}(x) - e_i^{(m)} W_{i-1}^{(m)}(x) \quad (1)$$

et

$$xW_i^{(m-1)}(x) = W_{i+1}^{(m)}(x) + q_{i+1}^{(m)} W_i^{(m)}(x) \quad (2)$$

que l'on appellera relation de récurrence sur deux antidiagonales adjacentes

et que les $(q_k^{(m)})$ et $(e_k^{(m)})$ satisfont les relations

$$q_{k+1}^{(m)} + e_{k+1}^{(m)} = q_{k+1}^{(m-1)} + e_k^{(m-1)} \quad (3)$$

et

$$e_k^{(m)} q_{k+1}^{(m)} = e_k^{(m-1)} q_k^{(m-1)} \quad (4)$$

Définition 7.2. : On appelle algorithme $\tilde{q}d$ l'algorithme qui calcule les coefficients $(q_k^{(m)})$ et $(e_k^{(m)})$ qui interviennent dans les relations de récurrence sur deux antidiagonales adjacentes.

On rangera les éléments calculés avec l'algorithme $\tilde{q}d$ dans un tableau que l'on appellera table $\tilde{q}d$, ce tableau est de la forme :

$$\begin{array}{ccccccc}
 \gamma_{q_0}^{(0)} & & \gamma_{q_1}^{(1)} & & & & \\
 & \gamma_{e_0}^{(1)} & & \gamma_{e_1}^{(2)} & & & \\
 \gamma_{q_0}^{(1)} & & \gamma_{q_1}^{(2)} & & \gamma_{q_2}^{(3)} & & \\
 & \gamma_{e_0}^{(2)} & & \gamma_{e_1}^{(3)} & & \gamma_{e_2}^{(4)} & \\
 \gamma_{q_0}^{(2)} & \vdots & \gamma_{q_1}^{(3)} & \vdots & \gamma_{q_2}^{(4)} & \vdots & \gamma_{q_3}^{(5)} \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
 \end{array}$$

l'indice supérieure étant l'indice de l'antidiagonale et l'autre celui de la colonne.

Grâce à la propriété 4.13

$$\gamma^{(m)}(x W_k^{(m)}(x)) = \gamma^{(m-1)}(W_k^{(m)}(x))$$

de [1, page 316] toutes les relations entre les polynômes orthogonaux et les éléments de la table q_d s'étendent au cas des polynômes $W_k^{(m)}(x)$ et des éléments de la table $\tilde{q}_d$. Il suffit de transformer tous les passages de l'indice n à $n+1$ des diagonales en passage de l'indice m à $m-1$ des antidiagonales.

On obtient par exemple la généralisation des relations (1) et (2) :

- si $W_i^{(m-1)}$ est O.R.

$$(5) \quad W_i^{(m-1)}(x) = \gamma_{i-\text{pr}(i+1,m)}^{(m)}(x) W_{\tilde{\text{pr}}(i+1,m)}^{(m)}(x) + \gamma_{i+1}^{(m)} W_{\tilde{\text{pr}}(\text{pr}(i+1,m),m-1)}^{(m-1)}(x)$$

et si de plus $W_{i+1}^{(m)}$ est O.R.

$$(6) \quad x W_i^{(m-1)}(x) = W_{i+1}^{(m)}(x) + \gamma_{i+1}^{(m)} W_{\tilde{\text{pr}}(i+1,m)}^{(m)}(x)$$

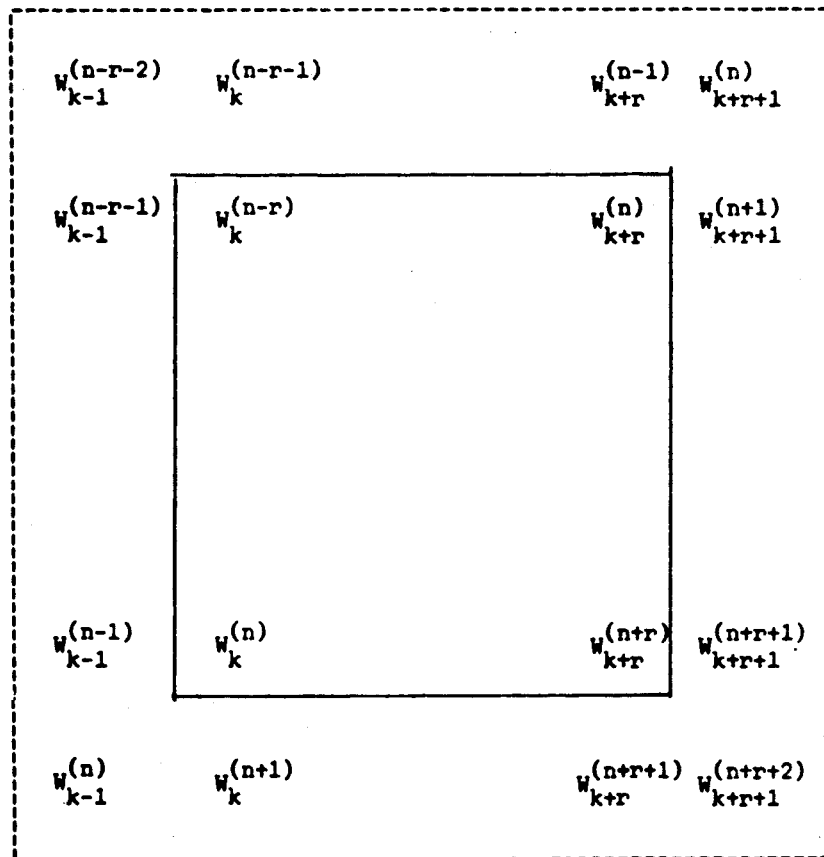
$$\text{avec } \gamma_{i+1}^{(m)} = \begin{cases} -e_i^{(m)} & \text{si } w_{i-1}^{(m-1)} \text{ est O.R.} \\ \gamma_{i+1}^{(m)} & \text{sinon} \end{cases}$$

Et les relations rhomboïdales classiques donnent les relations (3) et (4), avec au départ

$$e_0^{(m)} = \gamma_0^{(m)} = 0 \text{ et } q_1^{(m)} = \frac{c_{m-1}}{c_1} \text{ si } c_m \neq 0.$$

Dans le cas d'une table W non normale, l'ensemble des résultats pour le contournement des blocs P s'applique aux blocs W. En effet on obtient les relations "spéciales" pour contourner les blocs W en modifiant les indices supérieurs des relations de contournement des blocs P, de façon symétrique par rapport à l'antidiagonale principale du bloc.

Exemple : Si on a un bloc (n, k, r)



où n est l'indice de l'antidiagonale principale,
 r telle que $r+1$ est la largeur du bloc
et k est l'indice de la première colonne du bloc,

alors $n+r+1-i \xrightarrow{0} n-r-1+i$

Montrons qu'il existe une relation entre les algorithmes $\tilde{q}d$ et qd .

Considérons une suite de nombres réels $(c_i)_{i \in \mathbb{N}}$ et définissons les fonctionnelles linéaires $\gamma^{(m)}$ et $b^{(m)}$ telles que pour N fixé

$$\begin{aligned} \gamma^{(N-m)}(x^j) &= c_{N-m-j} \\ \forall m \in \mathbb{N} \quad & \text{pour } j \in \mathbb{N}, N-m-j \geq 0 \\ b^{(m)}(x^j) &= b_{m+j} = c_{N-m-j} \end{aligned}$$

On remarque que $b^{(m)} = \gamma^{(N-m)} \quad \forall m \in \mathbb{N}$ et que

$$H_{k+1}(b_m) = H_{k+1}(c_{N-m-2k})$$

Corollaire 7.3. : γ et b ont les mêmes blocs H .

Notons $\bar{P}_k^{(m)}(x)$ (resp. $w_k^{(m)}(x)$) les polynômes orthogonaux par rapport aux $b^{(m)}$ (resp. $\gamma^{(m)}$).

Corollaire 7.4. : Si $\bar{P}_k^{(m)}(x)$ est O.R. alors $w_k^{(N-m)}(x)$ est O.R. avec k et m tels que $0 \leq m \leq N-2k+1$.

Corollaire 7.5. : Si $\bar{P}_{k+1}^{(m)}(x)$ est O.R. alors $\bar{P}_{pr(k+1,m)}^{(m)}(x) = w_{\tilde{\tilde{pr(k+1,N-m)}}}^{(N-m)}(x)$

Ce corollaire permet d'écrire $\bar{P}_{pr(pr(k+1,m),m)}^{(m)}(x) = w_{\tilde{\tilde{pr(pr(k+1,N-m),N-m)}}}^{(N-m)}(x)$

$$\text{et } \bar{P}_{pr(pr(k+1,m),m+1)}^{(m+1)}(x) = w_{\tilde{\tilde{pr(pr(k+1,N-m),N-m-1)}}}^{(N-m-1)}(x)$$

Corollaire 7.6. : Si $\overline{P}_k^{(m)}(x)$ est O.R. alors $\gamma_{k+1}^{(N-m)} = E_{k+1}^{(m)}$ c'est-à-dire

$$e_k^{(N-m)} = e_k^{(m)} \text{ si } \overline{P}_{k-1}^{(m+1)} \text{ est O.R.}$$

$$c_{k+1}^{(N-m)} = c_{k+1}^{(m)} \text{ sinon}$$

avec k et m tels que $0 \leq m \leq N-2k$

et si de plus $\overline{P}_{k-1}^{(m+1)}$ est O.R. on a $\gamma_k^{(N-m)} = q_k^{(m)}$ avec m et k
tels que $0 \leq m \leq N-2k+1$

Ce corollaire se démontre en remplaçant les polynômes $\overline{P}_k^{(m)}(x)$ par les polynômes $W_k^{(N-m)}(x)$ dans les relations de récurrence sur deux diagonales adjacentes et en identifiant les relations obtenues avec les relations (5) et (6).

Ce corollaire montre que le comportement de l'algorithme $\tilde{q}d$ est symétrique, par rapport à une horizontale, de celui de l'algorithme qd .

CONCLUSION : Appliquer l'algorithme $\tilde{q}d$ à une suite (c_j) $0 \leq j \leq N$ revient à appliquer l'algorithme qd à la suite $(b_j = c_{N-j})$ $0 \leq j \leq N$, changer les indices m des diagonales en $N-m$ et faire une symétrie par rapport à une horizontale.

Donnons un algorithme de passage de la table $qd(b_j)$ à la table $\tilde{q}d(c_j)$.

Algorithme

Les $\overline{P}_k^{(m)}$ sont les polynômes orthogonaux par rapport aux $b^{(m)}$.

Les $W_k^{(m)}$ sont les polynômes orthogonaux par rapport aux $\gamma^{(m)}$.

On notera $q_k^{(m)}$ et $e_k^{(m)}$ les éléments de la table $qd(b_j)$ et, $\gamma_k^{(m)}$ et $e_k^{(m)}$ ceux de la table $\tilde{q}d(c_j)$.

Le passage d'un numéro au suivant se fait en séquence sauf s'il existe un ordre aller en.

On pose $j = 0$ et $k = 1$

- 1 - Si $\overline{P}_k^{(j)}$ est non O.R. alors $w_k^{(N-j)}$ est non O.R, aller en 4
sinon $w_k^{(N-j)}$ est O.R.
- 2 - Si $\overline{P}_{k-1}^{(j+1)}$ est non O.R, c'est-à-dire si $\overline{P}_k^{(j)}$ est à l'Est d'un bloc et donc $\gamma_k^{(j)}$ est infini, alors $w_k^{(N-j)}$ est à l'Est d'un bloc W, aller en 3.

$$\gamma_k^{(N-j)} = q_k^{(j)}$$

si $j \geq N-2k+1$ aller en 5.

- 3 - Si $\overline{P}_k^{(j+1)}$ est non O.R, c'est-à-dire si $\overline{P}_k^{(j)}$ est au Nord d'un bloc P et donc $e_k^{(j)}$ est infini ou indéfini, alors $w_k^{(N-j)}$ est au Sud d'un bloc W, aller en 4.

$$e_k^{(N-j)} = e_k^{(j)} \text{ si } \overline{P}_{k-1}^{(j+1)} \text{ est non O.R.}$$

$$c_{k+1}^{(N-j)} = c_{k+1}^{(j)} \text{ sinon.}$$

- 4 - Si $j < N-2k+1$, $j+1 \rightarrow j$ et aller en 1.

- 5 - Si $N-2k+1 \leq 0$ fin

$k+1 \rightarrow k$, $j = 0$ et aller en 1.

II. MISE EN OEUVRE

Elle se fait avec l'ensemble des sous-programmes de calcul de la table qd et le sous-programme QDTILD.

Mode d'Emploi du sous-programme QDTILD

Appel : CALL QDTILD(C, N)

Paramètres : C est le tableau des coefficients de la série formelle, il est indicé de 0 à N.

C et N sont des paramètres d'entrée.

SOUS-PROGRAMMES UTILISES : QUDE, DETECT et EST.

Description : Ce sous-programme calcule la table qd des polynômes orthogonaux satisfaisant la relation de récurrence à trois termes sur les anti-diagonales.

Ce sous-programme donne aussi l'allure de la table de ces polynômes orthogonaux.

EXEMPLE D'APPEL DES SOUS-PROGRAMMES

On présente un programme qui lit NL et les coefficients c_i de la série formelle, et appelle le sous-programme QDTILD, ce dernier utilise QUDE, EST et DETECT.

```

1.  C*****
2.  C    PROGRAMME PRINCIPAL POUR LE CALCUL DE LA TABLE Q-D A PARTIR DE LA
3.  C    RELATION DE RECURRENCE A TROIS TERMES SUR LES ANTIDIAGONALES
4.  C    C EST LE TABLEAU DES COEFFICIENTS DE LA SERIE FORMELLE, INDICE DE ZERO
5.  C    A NL
6.  C*****
7.  C*****CARTES POUVANT ETRE MODIFIEES PAR L'UTILISATEUR*****
8.  C    COMMON/CK(-20:20,0:20),DC(-20:20),DAR(-20:20,0:20),DD(-20:20,0:20)
9.  C    1,DE(-20:20,0:20),NDF(-20:20,0:20)/S/N(20),K(20),NR(20)
10. C    DIMENSION T(0:20)
11. C    LGB=105
12. C*****
13. C    READ(LGB,20)NL
14. C    READ(LGB,30)(T(I),I=0,NL)
15. C    CALL QDTILD(T,NL)
16. C    30 FORMAT(5F10.4)
17. C    20 FORMAT(I3)
18. C    STOP
19. C    END

```

Pour NL = 10 et les c_i suivants :

```

CALCUL DE LA TABLE Q-D INILDA POUR LES COEFFICIENTS:
C( 0)= -.10000000000000000+01
C( 1)= -.10000000000000000+01
C( 2)= .10000000000000000+01
C( 3)= .00000000000000000+00
C( 4)= .00000000000000000+00
C( 5)= .00000000000000000+00
C( 6)= .10000000000000000+01
C( 7)= .10000000000000000+01
C( 8)= .20000000000000000+01
C( 9)= -.20000000000000000+01
C(10)= .10000000000000000+01

```

On a les résultats suivants :

Le tableau

N= -1	+	+			
N= 0	+	+	+		
N= 1	+	0	0	0	
N= 2	+	0	0	0	+
N= 3	+	0	0	0	+
N= 4	+	+	+	+	+
N= 5	+	+	+	+	+
N= 6	+	+	+	+	
N= 7	+	+	+		
N= 8	+	+			
N= 9	+				
N= 10					

où les numéros de gauche sont les indices des antidiagonales, les + représentent les polynômes orthogonaux réguliers et les 0 les polynômes orthogonaux dans les blocs,

donne l'allure de la table W des polynômes orthogonaux $w_k^{(n)}$.

Les valeurs des éléments de la table $\tilde{q}d$ sont :



INDICE DE COLONNE= 1

VALEUR DE Q		VALEUR DE E	INDICE DE COLONNE= 2	
N=	1	.1000000000000000UD+01	VALEUR DE Q	VALEUR DE E
N= 2	-	.1000000000000000UD+01(*)	.2000000000000000UD+01	-.2000000000000000UD+01(*)
N= 3	INFINI OU INDEFINI	INFINI OU INDEFINI	INFINI OU INDEFINI	INFINI OU INDEFINI
N= 4	INFINI OU INDEFINI	INFINI OU INDEFINI	INFINI OU INDEFINI	INFINI OU INDEFINI
N= 5	INFINI OU INDEFINI	INFINI OU INDEFINI	INFINI OU INDEFINI	INFINI OU INDEFINI
N= 6	.0000000000000000UD+00(*)	INFINI OU INDEFINI	INFINI OU INDEFINI	INFINI OU INDEFINI
N= 7	.1000000000000000UD+01	-.1000000000000000UD+01	.0000000000000000UD+00(*)	.1000000000000000UD+01
N= 8	.5000000000000000UD+00	.5000000000000000UD+00	-.2000000000000000UD+01	-.1666666666666667D+01
N= 9	-.1000000000000000UD+01	.1500000000000000UD+01	.1666666666666667D+00	.3166666666666667D+01
N= 10	-.2000000000000000UD+01	.1000000000000000UD+01	-.1500000000000000UD+01	

INDICE DE COLONNE= 3

VALEUR DE Q		VALEUR DE E	INDICE DE COLONNE= 4	
N=	5	INFINI OU INDEFINI	VALEUR DE Q	VALEUR DE E
N= 6	INFINI OU INDEFINI	INFINI OU INDEFINI	INFINI OU INDEFINI	.1000000000000000UD+01(*)
N= 7	INFINI OU INDEFINI	INFINI OU INDEFINI	INFINI OU INDEFINI	-.6000000000000000UD+01
N= 8	.0000000000000000UD+00(*)	INFINI OU INDEFINI	.5000000000000000UD+01	.4166666666666667D+01
N= 9	.1200000000000000UD+01	-.2000000000000000UD+00	.6333333333333333D+00	
N= 10	-.8771929624561400-01	-.378947368421053D+00		

INDICE DE COLONNE= 5

VALEUR DE Q		VALEUR DE E
N= 9	.1666666666666667D+00	.1366666666666667D+01
N= 10	-.7200000000000000UD+01	

Les numéros de gauche sont les indices des antidiagonales.

(*) valeurs des éléments qui sont dans le bloc $\tilde{q}_d$.




```

1.      SUBROUTINE QUTILD(C,NL)
2.      C*****
3.      C      CALCUL DE LA TABLE Q-D DES POLYNOMES ORTHOGONAUX SATISFAISANT LA
4.      C      RELATION DE RECURRENCE A TROIS TERMES SUR LES ANTIDIAGONALES
5.      C      C EST LE TABLEAU DES COEFFICIENTS DE LA SERIE FORMELLE, INDICE DE ZERO
6.      C      A NL
7.      C      CE SOUS-PROGRAMME UTILISE LES SOUS-PROGRAMMES      QUDE
8.      C      EST
9.      C      DETECT
10.     C*****
11.     C*****CARTES POUVANT ETRE MODIFIEES PAR L'UTILISATEUR*****
12.     COMMON/DCR(-20:20,0:20),T(-20:20),DBR(-20:20,0:20),DD(-20:20,0:20),
13.     1DE(-20:20,0:20),NDF(-20:20,0:20)/S/N(20),K(20),NR(20)
14.     DIMENSIONC(NL),NF(-1:20,0:20),Q(N:20,0:20),E(N:20,0:20)
15.     MAT=108
16.     C*****
17.     WRITE(MAT,755)
18.     755 FORMAT(2A,'CALCUL DE LA TABLE Q-D THILDA POUR LES COEFFICIENTS:')
19.     756 FORMAT(//,2X,'CALCUL DE LA TABLE Q-D POUR LES COEFFICIENTS:')
20.     DO 1 I=NL,0,-1
21.     T(I)=C(NL-I);NF(I,0)=1;KA=NL-I;WRITE(MAT,20)KA,C(KA)
22.     1 CONTINUE;WRITE(MAT,756);CALL QUDE(NL,0,0,LA);L1=1;I=0
23.     WRITE(MAT,880)
24.     C*****
25.     C      SYMETRIE PAR RAPPORT A UNE HORIZONTALE
26.     C*****
27.     13 IF(NDF(J,L1).EQ.0)NF(NL-J,L1)=0;GO TO 12
28.     NF(NL-J,L1)=1;IF(NDF(J+1,L1-1).EQ.0)GO TO 11
29.     Q(NL-J,L1)=DQ(J,L1)
30.     11 IF(J.GT.NL-2*L1) GO TO 12
31.     IF(NDF(J+1,L1).GT.0)E(NL-J,L1)=DE(J,L1)
32.     12 IF(J.LT.NL-2*L1+1)J=J+1;GO TO 13
33.     IF(NL-2*L1+1.LE.0)J=1;I=1;GO TO 820
34.     L1=L1+1;J=0;GO TO 13
35.     C*****
36.     C      SORTIE DE LA TABLE Q-D
37.     C*****
38.     820 WRITE(MAT,780)I;WRITE(MAT,675)
39.     IF(NF(J,I).EQ.0)WRITE(MAT,850)J;GO TO 814
40.     IF(NF(J-1,I-1).EQ.0)GO TO 704
41.     WRITE(MAT,801)J,Q(J,I);GO TO 814
42.     806 IF(NF(J,I).EQ.0)WRITE(MAT,856);WRITE(MAT,850)J;GO TO 814
43.     IF(NF(J-1,I).EQ.0) GO TO 815
44.     WRITE(MAT,809)E(J,I);IF(NF(J-1,I-1).EQ.0)GO TO 704
45.     816 WRITE(MAT,801)J,Q(J,I);GO TO 814
46.     704 WRITE(MAT,850)J;GO TO 814
47.     815 WRITE(MAT,856);IF(NF(J-1,I-1).EQ.0)GO TO 704
48.     GO TO 816
49.     814 IF(J.LT.NL)J=J+1;GO TO 806
50.     IF(2*I.LT.NL)I=I+1;J=2*I-1;GO TO 820
51.     MOY=-1;I=0;NF(-1,0)=1
52.     C*****
53.     C      SORTIE DE L'ALLURE DE LA TABLE DES POLYNOMES ORTHOGONAUX
54.     C*****
55.     936 DO 48 J=MOY,NL+1
56.     IF(NF(J,I).EQ.1)NF(J,I)=1H;GO TO 48
57.     NF(J,I)=1H0
58.     48 CONTINUE;IF(MOY.LT.NL)MOY=MOY+2;I=I+1;GO TO 936
59.     ME=(NL+1)/2;KA=-2
60.     DO 39 J=0,ME-1;WRITE(MAT,1002);KA=KA+1
61.     39 WRITE(MAT,800)(KA,(NF(J+I,I),I=0,J+1))
62.     DO 38 J=0,NL-ME;KA=ME-1+J;WRITE(MAT,1002)
63.     38 WRITE(MAT,800)(KA,(NF(J+ME+I,I),I=0,NL-ME-J))
64.     WRITE(MAT,1002);WRITE(MAT,800)NL;RETURN
65.     625 FORMAT(//,8X,'VALEUR DE 0',17X,'VALEUR DE E',/)
66.     780 FORMAT(//,20X,'INDICE DE COLONNE=',I2,/)
67.     850 FORMAT(3X,'N=',I3,4X,'INFINI OU INDEFINI')
68.     856 FORMAT(3X,'INFINI OU INDEFINI')
69.     20 FORMAT(3X,'C('',I3,'')=',D22.15)
70.     1002 FORMAT(//)
71.     801 FORMAT(1X,'N=',I3,2X,D22.15,/)
72.     809 FORMAT(32X,D22.15)
73.     880 FORMAT(//,5X,'TABLE Q-D THILDA',/)
74.     800 FORMAT(1X,'N=',I3,5X,20(A1,5X))
75.     END

```



HUITIEME PARTIE

SOUS-PROGRAMMES ECRITS EN ARITHMETIQUE RATIONNELLE ETENDUE
VIII.1. INTRODUCTION

L'arithmétique rationnelle est une arithmétique dont les données associées aux opérateurs sont des nombres rationnels, elle est exacte si les données de départ sont des rationnels. C'est dans le but d'avoir des résultats exacts que nous reprenons l'essentiel des sous-programmes : calcul des éléments de la table qd ; calcul des polynômes orthogonaux et leurs associés ; calcul des approximants de Padé en un point et en deux points et le calcul des approximants des séries des fonctions. Evidemment l'arithmétique rationnelle entraîne la manipulation de nombres entiers qui dépassent très vite la capacité machine, nous avons essayé de contourner cette difficulté avec l'arithmétique rationnelle étendue [2]. Cette dernière écrit les entiers dans des tableaux et code sur ces tableaux les différents opérateurs de l'arithmétique rationnelle. Cette fois encore on a assez vite dépassé la capacité autorisée en travaillant en base 10^4 , la plus grande autorisée avec les sous-programmes de [2]. Nous avons donc été amenés à considérer une base plus importante (10^8) pour retarder le dépassement de capacité, et pour cela il fallait modifier, sinon reprendre les sous-programmes qui font intervenir la multiplication de deux entiers. Les sous-programmes MUL, MUL1, DIV1 et DIV2 ont été repris et quelques lignes de DIVE ont été modifiées.

On sait que quelle que soit la base B un entier N peut s'écrire sous la forme $N_1(N_2 B^{L-2} + N_3 B^{L-3} + \dots + N_{L-1} B + N_L)$ avec $N_1 = \text{signe de } N$ et les N_i tels que $N_i < B \quad \forall i$.

Dans le cas où B est une puissance de 10, $B = 10^k$, les N_i sont des entiers ayant au plus k chiffres.

Définition 8.1. : Le vecteur $(N_1, N_2, \dots, N_L)$ est appelé T.E.P. de longueur L et les N_i sont des mots.

MULTIPLICATION DE DEUX T.E.P.

Soient les T.E.P. $N = (N_1, N_2, \dots, N_L)$
et $M = (M_1, M_2, \dots, M_L)$

On sait que le produit de N et M est défini par les produits partiels $N_i M_j$. Dans un ordinateur la représentation des entiers est limitée par la capacité d'un mot mémoire ($2^{31} - 1$ sur l'IRIS 80) et donc en base 10^k , pour $k \geq 5$, dès que N_i et M_j sont très grands $N_i M_j$ dépasse cette capacité. Pour éviter le dépassement de capacité par $N_i M_j$, on peut écrire N_i et M_j sous forme de couple, de telle sorte que le produit de chaque élément du couple de N_i par n'importe quel élément du couple de M_j soit autorisé, et faire convenablement la multiplication.

Pour $k = 8$

On peut écrire N_i et M_j en base 10^4 :

$$N_i = N_{i1} \times 10^4 + N_{i2} = (N_{i1}, N_{i2})$$

$$M_j = M_{j1} \times 10^4 + M_{j2} = (M_{j1}, M_{j2})$$

et l'on a

$$\begin{aligned} N_i M_j &= (N_{i1}, N_{i2}) \times (M_{j1}, M_{j2}) = N_{i1} \times M_{j1} \cdot 10^8 + (N_{i2} \times M_{j1} + N_{i1} \times M_{j2}) \cdot 10^4 \\ &\quad + N_{i2} \times M_{j2} \end{aligned}$$

Et donc le produit $P = (N_2, \dots, N_L) \times M_j = (P_{j2}, \dots, P_{jL})$

$$\text{avec } P_{jL} = (N_{L2} \times M_{j1} + N_{L1} \times M_{j2}) \cdot 10^4 + M_{j2} \times N_{L2}$$

$$\text{et } P_{ji} = (N_{i2} \times M_{j1} + N_{i1} \times M_{j2}) \cdot 10^4 + N_{i2} \times M_{j2} + N_{(i+1)1} \times$$

avec $i = 2, \dots, L-1$.

Le produit de deux T.E.P. se fait comme dans [2, page 40] avec les P_{ij} que nous venons de définir.

DIVISION ENTIERE D'UN T.E.P.

PAR UN ENTIER INFERIEUR A LA BASE

Exemple : Considérons la division entière du T.E.P. $(N_1, N_2, \dots, N_6)$ en base 10^6 , avec $N_2 = N_3 = N_4 = 0$; $N_5 = 200000$ et $N_6 = 1$, par l'entier $I = 400000$.

$$Q_6 = N_6/I = 0$$

$$r = \text{mod}(N_6, I) = 1$$

$N_5 \cdot 10^6 + r = 200000 \times 10^6 + 1 = 200000000001 > 2^{31} - 1$, on a donc dépassement de la capacité machine.

Pour éviter le dépassement de la capacité machine on cherche d'abord le premier mot non nul, soit N_s ce mot. On pose

$$Q_s = N_s/I \quad \text{et} \quad r = \text{mod}(N_s, I).$$

Pour calculer les autres Q_j , $j > s$, on fait la division comme à la main en abaissant chiffre après chiffre le mot N_j sur le dernier reste de la division entière par I , chaque fois que l'on abaisse un chiffre on calcule le quotient et le nouveau reste. Quand on finit un mot, on continue avec le reste et le mot suivant.

On notera $NQ(J, K, \cdot)$ (resp. $NT(J, \cdot)$) le T.E.P.

$(NQ(J, K, 1), NQ(J, K, 2), \dots, NQ(J, K, L))$
(resp. $(NT(J, 1), NT(J, 2), \dots, NT(J, L))$).

On mémorisera un entier ϕ_k^j à deux indices (resp. ϕ^j à un indice) dans un T.E.P. $PH(J, K, \cdot)$ (resp. $PH(J, \cdot)$), et donc en arithmétique rationnelle étendue un rationnel à deux indices ϕ_k^j (resp. un indice ϕ^j) s'écrira sous la forme

$$\phi_k^j = \text{NPH}(J, K, \cdot) / \text{LPH}(J, K, \cdot)$$

$$(\text{resp. } \phi^j = \text{NPH}(J, \cdot) / \text{LPH}(J, \cdot))$$

Les sous-programmes de [2], étant conçus pour les rapports des T.E.P. de la forme $N(\cdot)/L(\cdot)$, nous ont conduits à écrire un certain nombre de sous-programmes de passage de $\text{NQ}(J, K, \cdot)/\text{LQ}(J, K, \cdot)$ (resp. $\text{NC}(J, \cdot)/\text{LC}(J, \cdot)$) à $N(\cdot)/L(\cdot)$ pour pouvoir les utiliser. Le passage dans le sens contraire fait aussi l'objet des sous-programmes.

LES SOUS-PROGRAMMES DE TRANSFERT

* * * REPAR * * * *

Appel : CALL REPAR(J, K, NQ, LQ, L, NX, LX)

Paramètres : $\text{NQ}(J, K, \cdot)$, $\text{LQ}(J, K, \cdot)$, $\text{NX}(\cdot)$ et $\text{LX}(\cdot)$ sont des T.E.P. de longueur L.

J est un entier pouvant être négatif.

K entier toujours positif.

L, J, K, NQ et LQ sont des paramètres d'entrée et, NX et LX sont des paramètres de sortie.

Description : Ce sous-programme transfère respectivement les T.E.P. $\text{NQ}(J, K, \cdot)$ et $\text{LQ}(J, K, \cdot)$ dans les T.E.P. $\text{NX}(\cdot)$ et $\text{LX}(\cdot)$.

* * * RANGE * * * *

Appel : CALL RANGE(J, NC, LC, L, NX, LX)

Paramètres : $\text{NC}(J, \cdot)$, $\text{LC}(J, \cdot)$, $\text{NX}(\cdot)$ et $\text{LX}(\cdot)$ sont des T.E.P. de longueur L.

J est un entier qui peut être négatif.

J, NC, LC et L sont des paramètres d'entrée et, NX et LX sont des paramètres de sortie.

Description : Ce sous-programme transfère respectivement les T.E.P. NC(J, •) et LC(J, •) dans les T.E.P. NX(•) et LX(•).

* * * RANGE1 * * * *

Appel : CALL RANGE1(J, NC, LC, L, NX, LX)

Paramètres : Ce sont les mêmes que dans RANGE, mais avec J positif.

Description : Même description que RANGE.

* * * MACHIN * * * *

Appel : CALL MACHIN(J, K, NX, LX, L, NQ, LQ)

Paramètres : NX(•), LX(•), NQ(J, K, •) et LQ(J, K, •) sont des T.E.P. de longueur L.

J est un entier pouvant être négatif.

K un entier positif.

J, K, NX, LX et L sont des paramètres d'entrée et, NQ et LQ sont des paramètres de sortie.

Description : Ce sous-programme transfère respectivement les T.E.P. NX(•) et LX(•) dans les T.E.P. NQ(J, K, •) et LQ(J, K, •).

* * * MISE * * * *

Appel : CALL MISE(J, NX, LX, L, NC, LC)

Paramètres : NX(•), LX(•), NX(J, •) et LX(J, •) sont des T.E.P. de longueur L.

J est un entier pouvant prendre des valeurs négatives.

J, L, NX et LX sont des paramètres d'entrée et, NC et LC sont des paramètres de sortie.

Description : Ce sous-programme transfère respectivement les T.E.P. NX(•) et LX(•) dans les T.E.P. NC(J, •) et LC(J, •).

* * * MISE1 * * * *

Appel : CALL MISE1(J, NX, LX, L, NC, LC)

Paramètres : Les mêmes que dans MISE, mais avec J positif.

Description : Idem que pour MISE.

On a écrit un sous-programme de transfert de $\pm NQ(J, K, \bullet)/LQ(J, K, \bullet)$ dans $NB(J1, K1, \bullet)/LB(J1, K1, \bullet)$:

* * * EGAL * * * *

Appel : CALL EGAL(MA, J, K, NQ, LQ, L, J1, K1, NB, LB)

Paramètres : $NQ(J, K, \bullet)$, $LQ(J, K, \bullet)$, $NB(J1, K1, \bullet)$ et $LB(J1, K1, \bullet)$ sont des T.E.P. de longueur L.

J et J1 sont des entiers quelconques et, K1 et K sont des entiers positifs.

MA = +1 ou -1 selon le cas.

MA, J, K, J1, K1, L, NQ, LQ sont des paramètres d'entrée et, NB et LB sont des paramètres de sortie.

Description : Ce sous-programme transfère respectivement les T.E.P.

$MA \times NQ(J, K, \bullet)$ et $LQ(J, K, \bullet)$ dans les T.E.P. $NB(J1, K1, \bullet)$ et $LB(J1, K1, \bullet)$.

On peut donner deux autres sous-programmes qui ne sont pas de transfert de valeurs : QUOTIE et IZERO.

L'ordinateur travaille avec des mots de longueur limitée, donc le quotient de deux T.E.P. effectué dans l'arithmétique habituelle de cet ordinateur ne pourra donner plus de chiffres significatifs. De plus si le nombre de chiffres d'un des deux T.E.P. dépasse celui d'un mot il faudra faire appel à un sous-programme particulier.

Le sous-programme QUOTIE calcule une approximation du quotient de deux T.E.P. sur IRIS 80 travaillant avec 15 chiffres de la manière suivante :

Soit M un T.E.P., on note $\text{chif}(M)$ le nombre de chiffres dans le T.E.P. M et on définit $\tilde{\text{chif}}$ tel que

$$\tilde{\text{chif}}(M) = \max(0, \text{chif}(M) - 15)$$

Tout T.E.P. M peut donc s'écrire sous la forme $M = (\tilde{m} + \tilde{\Delta m}) 10^{\tilde{\text{chif}}(M)}$ avec $\tilde{m}$ un nombre entier ayant au plus 15 chiffres et $\tilde{\Delta m}$, un nombre décimal ayant $(\text{chif}(M) - 15)$ chiffres, tel que $0 \leq \tilde{\Delta m} < 1$.

On peut remarquer que si $\text{chif}(M) \leq 15$ on a $\tilde{\text{chif}}(M) = 0$, $\tilde{\Delta m} = 0$ et $M = \tilde{m}$.

Soient M et N deux T.E.P. avec $N \neq 0$, le quotient $Q = \frac{M}{N}$ peut s'écrire sous la forme

$$Q = \frac{m}{n} 10^{\tilde{\text{chif}}(M) - \tilde{\text{chif}}(N)}$$

$$\begin{aligned} \text{avec } m &= \tilde{m} + \tilde{\Delta m} \\ \text{et } n &= \tilde{n} + \tilde{\Delta n}. \end{aligned}$$

L'approximant du quotient $Q = \frac{M}{N}$ calculé par le sous-programme QUOTIE est égal à $\bar{Q} = \frac{\bar{m}}{\bar{n}} 10^{\tilde{\text{chif}}(M) - \tilde{\text{chif}}(N)}$ où $\bar{m}$ (resp. $\bar{n}$) est égal à $\tilde{m}$ (resp. $\tilde{n}$) augmenté d'une unité si $\tilde{\Delta m}$ (resp. $\tilde{\Delta n}$) $\geq 0,5$.

On peut donc écrire $m = \bar{m} + \Delta m$ et $n = \bar{n} + \Delta n$ avec $|\Delta m| \leq 0,5$ et $|\Delta n| \leq 0,5$.

Calculons les majorations de $|\Delta Q| = |Q - \bar{Q}|$ suivant les cas.

En effet si $Q = \frac{m}{n} 10^{\tilde{\text{chif}}(M) - \tilde{\text{chif}}(N)}$ on a

$$|\Delta Q| = \left| \Delta \left(\frac{m}{n} \right) \right| 10^{\tilde{\text{chif}}(M) - \tilde{\text{chif}}(N)}$$

avec $\left| \Delta\left(\frac{m}{n}\right) \right| \leq \frac{1}{|n|} (|\Delta m| + \left| \frac{\bar{m}}{\bar{n}} \right| |\Delta n|) + |\epsilon_{\Delta}|$ où ϵ_{Δ} est l'erreur sur l'opération,

elle est toujours majorée par la précision machine, 10^{-15} pour l'IRIS 80.

Convention : On assimilera n à $\bar{n}$ lorsque $\Delta n \neq 0$.

CONSIDERONS DIFFERENTS CAS POSSIBLES

1° Si $\text{chif}(M)$ et $\text{chif}(N)$ sont inférieurs à 15

alors $\hat{\text{chif}}(M) = \hat{\text{chif}}(N) = 0$ et $\Delta m = \Delta n = 0$, donc

$$\left| \Delta\left(\frac{m}{n}\right) \right| = |\epsilon_{\Delta}| \leq 10^{-15}$$

$$\text{et } |\Delta Q| \leq 10^{-15}.$$

2° Si $\text{chif}(M) > 15$ et $\text{chif}(N) \leq 15$

alors $\hat{\text{chif}}(N) = 0$ et $\Delta n = 0$, donc

$$\left| \Delta\left(\frac{m}{n}\right) \right| \leq \frac{|\Delta m|}{|\bar{n}|} + |\epsilon_{\Delta}|$$

$$\text{et comme } |\bar{n}| \geq 10^{\text{chif}(N)-1}$$

on a $\left| \Delta\left(\frac{m}{n}\right) \right| \leq 0,5 \times 10^{-\text{chif}(N)+1} + 10^{-15} \leq 0,6 \times 10^{-\text{chif}(N)+1}$
car $-\text{chif}(N)+1 \geq -14$,

$$\text{et donc } |\Delta Q| \leq 0,6 \times 10^{\text{chif}(M)-14-\text{chif}(N)}$$

$$\leq 0,6 \times 10^{\text{chif}(M)-15} \text{ puisque } \text{chif}(N) \geq 1.$$

3° Si $\text{chif}(M) \leq 15$ et $\text{chif}(N) > 15$

alors $\hat{\text{chif}}(M) = 0$ et $\Delta m = 0$, donc

$$\left| \Delta\left(\frac{m}{n}\right) \right| \leq \frac{1}{|\bar{n}|} \frac{|\bar{m}|}{|\bar{n}|} |\Delta n| + |\epsilon_{\Delta}| \text{ puisque l'on assimile } n \text{ à } \bar{n} \text{ et}$$

comme $|\bar{n}| \geq 10^{14}$ et $|\bar{m}| < 10^{\text{chif}(M)}$,

$$\begin{aligned} \text{on a } \left| \Delta\left(\frac{m}{n}\right) \right| &\leq \frac{1}{10^{14}} \times \frac{10^{\text{chif}(M)}}{10^{14}} \times 0,5 + 10^{-15} \\ &\leq 5 \times 10^{-19+\text{chif}(M)} + 10^{-15} \end{aligned}$$

$$\text{or } -19 - \text{chif}(M) \leq -14$$

$$\text{donc } \left| \Delta\left(\frac{m}{n}\right) \right| \leq 5 \times 10^{-14} + 10^{-15} = 5,1 \times 10^{-14}$$

$$\text{et } |\Delta Q| \leq 5,1 \times 10^{-\text{chif}(N)+1}.$$

4° Si $\text{chif}(M) > 15$ et $\text{chif}(N) > 15$

à $\bar{n}$ alors $\left| \Delta\left(\frac{m}{n}\right) \right| \leq \frac{1}{|\bar{n}|} (|\Delta m| + \left| \frac{\bar{m}}{\bar{n}} \right| |\Delta n|) + |\epsilon_A|$ puisque l'on assimile n

$$\text{et comme } \frac{1}{10} < \left| \frac{\bar{m}}{\bar{n}} \right| < 10 \text{ et } |\bar{n}| \geq 10^{14}$$

$$\begin{aligned} \text{on a donc } \left| \Delta\left(\frac{m}{n}\right) \right| &\leq \frac{1}{10^{14}} (0,5 + 10 \times 0,5) + 10^{-15} \\ &\leq 56 \times 10^{-15} \end{aligned}$$

$$\text{et } |\Delta Q| \leq 56 \times 10^{\text{chif}(M) - \text{chif}(N) - 15}.$$

* * * QUOTIE * * * *

Appel : CALL QUOTIE(M, N, L, Q)

Paramètres : M(•) et N(•) sont des T.E.P. de longueur L.

Q est un réel.

M, N et L sont des paramètres d'entrée et Q est un paramètre de sortie.

Description : Approximation du quotient $Q = \frac{M}{N}$ par la méthode que l'on vient de décrire.

* * * IZERO * * * *

Appel : IZERO(N, L)

Paramètres : N(•) est un T.E.P. de longueur L

N et L sont des paramètres d'entrée.

Description : Cette fonction est égale à zéro si $N(\bullet) = 0$, et à 2 sinon.

Les sous-programmes ADDFRA, SOUFRA, MULFRA, DIVFRA, ADD, RAT, ECR, SOUS, INI, PGCD, SOM, SOM1, DIF, COMP, DIF1, LOGBAS et NOMCHI de [2] ont été utilisés sans aucune modification.

Les sous-programmes MUL, MUL1, DIV1 et DIV2 ont été repris, mais les paramètres d'entrée et de sortie n'ont pas été déplacés, et donc l'appel de ces sous-programmes se fait comme dans [2], il en est de même pour DIVE qui a été modifié sur quelques lignes.

Le sous-programme STO a été repris et les paramètres ont changé de position.

* * * STO * * * *

Appel : CALL STO(K, L, N)

Paramètres : K est un entier et N(•) un T.E.P. de longueur L.

K et L sont des paramètres d'entrée et N un paramètre de sortie.

Description : Mémorisation de l'entier K dans le T.E.P. N(•).


```

1. SUBROUTINE EGAL(MA,J1,I1,NQ,LQ,L1,N1,K1,NT,LT)
2. C*****
3. C NQ(J1,I1,..),LQ(J1,I1,..),NT(N1,K1,..) ET LT(M1,K1,..) SONT DES TEP DE
4. C LONGUEUR -L1-
5. C MA=1,SI NQ ET NT SONT DE MEME SIGNE
6. C -1 SINON
7. C TRANSFERT DE NQ(J1,I1,..) DANS MA*NT(N1,K1,..)
8. C LQ(J1,I1,..) DANS LT(M1,K1,..)
9. C*****
10. C*****CARTES POUVANT ETRE CHANGEES PAR L'UTILISATEUR*****
11. C DIMENSIONNQ(-10:10,0:10,L1),LQ(-10:10,0:10,L1),NT(-10:10,0:10,L1),
12. C 1LT(-10:10,0:10,L1)
13. C*****FIN*****
14. C DO 1 I=L1,2,-1
15. C NT(N1,K1,I)=NQ(J1,I1,I);LT(N1,K1,I)=LQ(J1,I1,I)
16. C 1 CONTINUE;LT(N1,K1,I)=LQ(J1,I1,I);NT(N1,K1,I)=MA*NQ(J1,I1,I)
17. C RETURN
18. C END

```

```

1. SUBROUTINE REPAR(J,K,NQ,LQ,L,NX,LX)
2. C*****
3. C TRANSFERT DES TEP NQ(J,K,..) ET LQ(J,K,..) DE LONGUEUR L DANS LES TEP
4. C NX ET LX DE LONGUEUR L
5. C*****
6. C*****CARTE POUVANT ETRE MODIFIEE PAR L'UTILISATEUR*****
7. C DIMENSIONNQ(-10:10,0:10,L),LQ(-10:10,0:10,L),NX(L),LX(L)
8. C*****FIN*****
9. C DO 1 I=1,L
10. C NX(I)=NQ(J,K,I);LX(I)=LQ(J,K,I)
11. C 1 CONTINUE;RETURN
12. C END

```

```

1. SUBROUTINE RANGE1(J,N,M,L,NX,LX)
2. C*****
3. C TRANSFERT DES TEP N(J,..) ET L(J,..) DE LONGUEUR L DANS LES TEP NX ET LX
4. C DE LONGUEUR L
5. C*****
6. C*****CARTE POUVANT ETRE MODIFIEE PAR L'UTILISATEUR*****
7. C DIMENSIONN(0:20,L),M(0:20,L),NX(L),LX(L)
8. C*****FIN*****
9. C DO 1 I=1,L
10. C NX(I)=N(J,I);LX(I)=M(J,I)
11. C 1 CONTINUE;RETURN
12. C END

```

```

1. SUBROUTINE MISE1(J,N,M,L,NT,LT)
2. C*****
3. C -N,-M,-NT(J,..) ET LT(J,..) SONT DES TEP DE LONGUEUR -L-
4. C TRANSFERT DES TEP -N- ET -M- DANS LES TEP NT(J,..) ET LT(J,..)
5. C*****
6. C*****CARTE POUVANT ETRE MODIFIEE PAR L'UTILISATEUR*****
7. C DIMENSIONNT(0:20,L),LT(0:20,L),N(L),M(L)
8. C*****FIN*****
9. C DO 1 I=1,L
10. C NT(J,I)=N(I)
11. C 1 LT(J,I)=M(I);RETURN
12. C END

```



```

1.      SUBROUTINE MISE(J,N,M,L,NT,LT)
2.      C*****
3.      C      -N,-M,-NT(J,..) ET LT(J,..) SONT DES TEP DE LONGUEUR -L-
4.      C      TRANSFERT DES TEP -N- ET -M- DANS LES TEP NT(J,..) ET LT(J,..)
5.      C*****
6.      C*****CARTE POUVANT ETRE MODIFIEE PAP L'UTILISATEUR*****
7.      C      DIMENSIONNT(-10:10,L),LT(-10:10,L),N(L),M(L)
8.      C*****FIN*****
9.      DO 1 I=1,L
10.         NT(J,I)=N(I)
11.         1 LT(J,I)=M(I);RETURN
12.      END

```

```

1.      SUBROUTINE RANGE(J,N,M,L,NX,LX)
2.      C*****
3.      C      TRANSFERT DES TEP N(J,..) ET L(J,..) DE LONGUEUR L DANS LES TEP NX ET LX
4.      C      DE LONGUEUR L
5.      C*****
6.      C*****CARTE POUVANT ETRE MODIFIEE PAR L'UTILISATEUR*****
7.      C      DIMENSIONN(-10:10,L),M(-10:10,L),NX(L),LX(L)
8.      C*****FIN*****
9.      DO 1 I=1,L
10.         NX(I)=N(J,I);LX(I)=M(J,I)
11.         1 CONTINUE;RETURN
12.      END

```

```

1.      SUBROUTINE MACHIN(J,K,N,M,L,NQ,LQ)
2.      C*****
3.      C      N,-M,-NQ(J,K,..) ET -LQ(J,K,..) SONT DES TEP DE LONGUEUR -L-
4.      C      TRANSFERT DES TEP N DANS NQ(J,K,..)
5.      C      M DANS LQ(J,K,..)
6.      C*****
7.      C*****CARTES POUVANT ETRE CHANGEES PAR L'UTILISATEUR*****
8.      C      DIMENSIONN(L),N(L),NQ(-10:10,0:10,L),LQ(-10:10,0:10,L)
9.      C*****FIN*****
10.     DO 1 I=1,L
11.        NQ(J,K,I)=N(I)
12.        1 LQ(J,K,I)=M(I);RETURN
13.     END

```

```

1.      SUBROUTINE DIV1(N,IA,IC,IR,L)
2.      C*****
3.      C      DIVISION ENTIERE D'UN TEP DE LONGUEUR L PAR UN ENTIER IA
4.      C      LE RESULTAT EST COMME DANS LE TEP IC, LE RESTE EST DANS LE TEP IR
5.      C*****
6.      COMMON/BLOCK1/IBASE
7.      DIMENSIONN(L),IC(L),IR(L)
8.      DO 3 I=1,L
9.         0 IC(I)=0;IC(1)=1;IB=0;I2=2
10.         4 DO 1 I=I2,L
11.            IF(N(I).NE.0)I1=I;GO TO 15
12.            1 CONTINUE;GO TO 11
13.            15 IB=-MOD(N(I1),IA);IC(I1)=N(I1)/IA;IF(I1.EQ.L)GO TO 11
14.            DO 3 I=I1+1,L
15.               IF(IB.EQ.0)I2=I;GO TO 4
16.               IBA=IBASE;I3=1;I2=N(I);IBAS=IBA
17.               12 IBA=IBA/10;IP=IB+10*I3+I2/IBA;IF(IP.GE.1A)GO TO 13
18.               IF(IBA.LE.1)IR=IP;GO TO 3
19.               I3=I3+1;GO TO 12
20.               15 IC(I)=(IP/IA)*(IBAS/10*I3)+IC(I);IB=MOD(IP,IA)
21.               IZ=MOD(N(I),IBA);IF(IBA.GT.1)I3=1;IBAS=IBA;GO TO 12
22.               3 CONTINUE
23.               11 DO 7 I=2,L-1
24.                  7 IR(I)=0;IR(1)=1;IR(L)=IB;RETURN
25.      END

```

```

1.      SUBROUTINE MUL1(IB,IC,IQ,K2,K3,L)
2.      C*****
3.      C      MULTIPLICATION D'UN TEP IB DE LONGUEUR L DONT LE PREMIER MOT NON
4.      C      NUL EST A L'INDICE K2 PAR UN TEP DONT SEUL LE MOT IC D'INDICE K3 EST
5.      C      NON NUL
6.      C      LE RESULTAT EST DANS IQ
7.      C*****
8.      COMMON/BLOK1/IBASE,L4,IN
9.      DIMENSION IB(L),IQ(L)
10.     IF(L+2.GT,K2+K3)GO TO 5
11.     DO 25 I=2,L
12.     25 IQ(I)=0;LA1=IC/10000;LA2=MOD(IC,10000);K=K3
13.     DO 1 I=L,K2,-1
14.     LB1=IB(I)/10000;LB2=MOD(IB(I),10000);IX=LB1*LA2+LA1*LB2
15.     IP=LA2*LB2+10000*MOD(IX,10000)
16.     IR=LA1*LB1+(IP/IBASE)+(IX/10000);IQ(K)=IQ(K)+MOD(IP,IBASE)
17.     IF(K.EQ,2)GO TO 6
18.     IQ(K-1)=IR+IQ(K-1)
19.     1 K=K-1;IP=0
20.     15 DO 10 J=K3,2,-1
21.     IA=(IQ(J)+IP)/IBASE;IQ(J)=MOD(IQ(J)+IP,IBASE);IP=IA
22.     10 CONTINUE;IF(IA.GT,0) GO TO 5
23.     RETURN
24.     6 IF(IR.EQ,0)IP=0;GO TO 15
25.     5 WRITE(IN,100);STOP
26.     100 FORMAT(3X,'DEPASSEMENT DE CAPACITE DANS MUL1')
27.     END

```

```

1.      SUBROUTINE DIV2(IX,IA,L)
2.      C*****
3.      C      DIVISION D UN TEP IX DE LONGUEUR L PAR UN ENTIER IA EGAL A UNE PUISSANCE
4.      C      DE 10 ET INFÉRIEUR A IBASE
5.      C*****
6.      COMMON/BLOK1/IBASE
7.      DIMENSION IX(L)
8.      IR=0
9.      DO 1 I=2,L
10.     IC=MOD(IX(I),IA)
11.     IX(I)=IR*(IBASE/IA)+(IX(I)/IA)
12.     1 IR=IC;RETURN
13.     END

```



```

1.      SUBROUTINE MUL(IA,IB,IC,L,ITA)
2.      C*****
3.      C      MULTIPLICATION DE DEUX TEP IA ET IB DE LONGUEUR L
4.      C      LE PRODUIT DANS LE TEP IC
5.      C      ITA=1,S'IL Y A DEBORDEMENT DANS LE RESULTAT IC
6.      C      =0,SINON
7.      C*****
8.      COMMON/BLOK1/IBASE,LM,IN
9.      DIMENSIONIA(L),IB(L),IC(L)
10.     ITA=0;IC(1)=IA(1)*IB(1)
11.     DO 1 I=2,L
12.     1 IC(I)=0
13.     DO 9 J=2,L
14.     IF(IA(J).NE.0)I1=J;GO TO 10
15.     9 CONTINUE;IC(1)=1;RETURN
16.     DO 12 J=2,L
17.     IF(IB(J).NE.0)I2=J;GO TO 13
18.     12 CONTINUE;IC(1)=1;RETURN
19.     13 IF(L+2.GT.I1+I2)GO TO 5
20.     DO 2 I=L,I2,-1
21.     IR=0;IF(IR(I).EQ.0) GO TO 2
22.     LR1=IR(I)/10000;LR2=MOD(IR(I),10000);K=0
23.     DO 3 J=L,I1,-1
24.     LA1=IA(J)/10000;LA2=MOD(IA(J),10000);IZ=LA2*LB1+LA1*LB2
25.     IP=IR+LA2*LB2+10000*MOD(IZ,10000);IC(I-K)=MOD(IP,IBASE)+IC(I-K)
26.     IR=LA1*LB1+(IP/IBASE)+(IZ/10000);IF(I-K.EQ.2) GO TO 6
27.     IF(J.EQ.I1)IC(I-K-1)=IR+IC(I-K-1)
28.     3 K=K+1
29.     2 CONTINUE;IP=0
30.     DO 14 J=L,2,-1
31.     ID=(IC(J)+IP)/IBASE;IC(J)=MOD(IC(J)+IP,IBASE);IP=ID
32.     14 CONTINUE;IF(IP.GT.0)GO TO 5
33.     RETURN
34.     6 IF(IR.EQ.0)IP=0;GO TO 15
35.     5 WRITE(IN,7);ITA=1;STOP
36.     7 FORMAT(2X,'DEPASSEMENT DE CAPACITE DANS LA MULTIPLICATION')
37.     END

```

```

1.      SUBROUTINE STO(K,L,M)
2.      C*****
3.      C      MEMORISATION D'UN ENTIER K DANS LE TEP -N- DE LONGUEUR L
4.      C*****
5.      COMMON/BLOK1/IBASE,LM,IN
6.      DIMENSIONM(L)
7.      N=K;M(1)=1;IF(N.LT.0)N=-N;M(1)=-1
8.      DO 2 I=L,2,-1
9.      M(I)=MOD(N,IBASE);N=N/IBASE
10.     2 CONTINUE;IF(N.GT.0)WRITE(IN,200)K;STOP
11.     200 FORMAT('LA VALEUR DE K TROP PETITE POUR LE NOMBRE ',022.15)
12.     RETURN
13.     END

```



```

1.      SUBROUTINE DIVE(A,C,Q,R,Q1,R1,L)
2.      C*****
3.      C** DIVISION ENTIERE D'UN TEP -A- PAR UN TEP -C- DE LONGUEUR -L- LE RESULTAT EST
4.      C** DONNE PAR LES TEP -Q,P- SOUS LA FORME (A=C*Q+R).
5.      C** LES TABLEAUX -Q1- ET -R1- DE LONGUEUR -L- SONT DES TABLEAUX DE CALCUL
6.      C** INTERMEDIAIRE.
7.      C** ALGORITHME : ON ISOLE LES DEUX PREMIERS MOTS NON NULS -C1- ET -C2- DU POIDS
8.      C** LE PLUS FORT DE -A- ET -C-. ON CALCULE C1=C1/C2. ON FORME LE QUOTIENT Q=C1+C1
9.      C** ON CALCULE Q1=Q*C PUIS R1=A-Q1, ET TANT QUE R1>C ON REMPLACE -A- PAR -R1- ET
10.     C** ON RECOMMENCE, SINON ON A R1=R ET ON A A=C*Q+R.
11.     C*****
12.     IMPLICIT INTEGER (A,C-Z), LOGICAL(B)
13.     DIMENSION A(L),C(L),Q(L),R(L),Q1(L),R1(L)
14.     COMMON/PL0K1/IBASE,LM,IN
15.     C*****
16.     C** RECHERCHE DU PREMIER MOT NON NUL DU DIVISEUR -C-.
17.     C*****
18.     DO 3 I=2,L
19.       IF(C(I).NE.0) GO TO 4
20.     3   CONTINUE
21.     C*****
22.     C** CAS OU LE DIVISEUR -C- EST NUL. ARRET DES CALCULS.
23.     C*****
24.     WRITE(IN,25)
25.     25  FORMAT('ERREUR DANS LA PROCEDURE''DIVE'',LE DENOMINATEUR EST NUL')
26.     STOP
27.     C*****
28.     C** DANS LE CAS OU -A- ET -C- NE SONT PAS DE MEME SIGNE, LE CALCUL EST ARRETE ET
29.     C** UN MESSAGE D'ERREUR EST IMPRIME.
30.     C*****
31.     4   K2=I
32.     IS=A(I)*C(1)
33.     IF(IS.GE.0) GO TO 28
34.     WRITE(IN,29)
35.     29  FORMAT('ERREUR:LA PROCEDURE''DIVE''N''ACCEPTTE QUE DES ENTIERS DE
36.     36  MEME SIGNE')
37.     STOP
38.     C*****
39.     C** -A- ET -C- ETANT DE MEME SIGNE ON LES CONSIDERE POSITIFS, ET ON LES COMPARE
40.     C** EN VALEUR ABSOLUE.
41.     C*****
42.     28  A(1)=1
43.     C(1)=1
44.     CALL COMP(A,C,L,N)
45.     C*****
46.     C** SI -A- EST PLUS PETIT QUE -C- EN VALEUR ABSOLUE, DEUX CAS : SI -A- EST NUL
47.     C** ALORS Q=R=0, SINON ON IMPRIME UN MESSAGE D'ERREUR.
48.     C*****
49.     GO TO (40,41,42) N
50.     42  DO 52 I=2,L
51.       IF(A(I).NE.0) GO TO 53
52.     52  CONTINUE
53.     DO 54 I=2,L
54.       Q(I)=0
55.       R(I)=0
56.       Q(1)=1
57.       R(1)=1
58.       RETURN
59.     53  WRITE(IN,43)
60.     43  FORMAT('ERREUR DANS LA PROCEDURE -DIV- LE DENOMINATEUR EST SUPERIE
61.     61  SUP AU NUMERATEUR')
62.     STOP
63.     C*****
64.     C** SI A=C ALORS R=0 ET Q=1.
65.     C*****
66.     40  DO 44 I=2,L
67.       Q(I)=0
68.     44  R(I)=0
69.       Q(1)=1
70.       R(1)=1
71.       Q(L)=1
72.       RETURN
73.     C*****
74.     C** SI -A- EST PLUS GRAND QUE -C- LE RESTE DU SOUS PROGRAMME VA FAIRE LA DIVISION
75.     C** ON ETUDIE EN PREMIER LE CAS OU LA VALEUR DE -C- EST ECRITE DANS UN SEUL MOT
76.     C** DANS CE CAS ON APPELLE LE SOUS-PROGRAMME (DIV1).
77.     C*****
78.     41  C2=C(I)
79.     BOUT=.FALSE.
80.     IF(K2.LT.L) GO TO 30
81.     CALL DIV1(A,C2,Q,R,L)
82.     RETURN
83.     C *****
84.     C** INITIALISATION DU RESTE INTERMEDIAIRE -R1- A -A-.
85.     C *****
86.     30  DO 1 I=2,L
87.       1  R1(I)=A(I)
88.       R1(1)=1

```

BUS
LILLE

```

89. C*****
90. C** INITIALISATION DU QUOTIENT=Q-ET DU QUOTIENT INTERMEDIAIRE-Q1-A ZERU.LE SIGNE
91. C** DE -Q1-DEPEND DE CELUI DE-R1-.
92. C*****
93. DO 2 I=2,L
94. 2 Q(I)=0
95. Q(1)=1
96. 5 DO 11 I=2,L
97. 11 Q1(I)=0
98. Q1(1)=1
99. IF(R1(1).GE.0) Q1(1)=-1
100. C*****
101. C** RECHERCHE DU PREMIER ELEMENT NON NUL DU DIVIDENDE A,ON LE MET DANS-C1,-R1-
102. C** ETANT SON INDICE.
103. C*****
104. DO 12 I=2,L
105. IF(R1(I).NE.0) GO TO 13
106. 12 CONTINUE
107. C*****
108. C**DIVISION DU PREMIER ELEMENT NON NUL DE -A- PAR LE PREMIER ELEMENT NON
109. C**NUL DE -C-.
110. C*****
111. 13 K1=I;C1=R1(I);IF(C1.GE.C2)C1=C1/C2;GO TO 14
112. IBA=IBASE;IRAS=IBA;I3=1;IZ=C1;C1=0
113. 61 IBA=IRAS/10**I3;IP=IZ*10**I3;IF(IP.GE.C2)GO TO 60
114. IF(THA.LE.1)GO TO 62
115. I3=I3+1;GO TO 61
116. 60 C1=C1+(IP/C2)*(IBAS/10**I3);IZ=MOD(IP,C2)
117. IF(ISA.GT.1)I3=1;ILAS=IBA;GO TO 61
118. 62 K1=K1+1
119. 14 K3=L-K2+K1
120. C*****
121. C** CALCUL DE Q1=C1*C PAR LA PROCEDURE(MUL1).
122. C*****
123. CALL MUL1(C,C1,Q1,K2,K3,L)
124. C*****
125. C** ON CALCUL LE QUOTIENT=Q-EN LUI AJOUTANT OU EN LUI RETRANCHANT-C1-SUIVANT QUE
126. C** -R1- DE L'ETAPE PRECEDANTE EST POSITIF OU NEGATIF.
127. C*****
128. IF(ROUL) GO TO 15
129. CALL SOM1(Q,C1,K3,L)
130. GO TO 17
131. 15 CALL DIF1(Q,C1,K3,L)
132. C*****
133. C**CALCUL DU RESTE
134. C*****
135. 17 CALL ADD(R1,Q1,R,L,ITE)
136. C*****
137. C** CRITERES D'ARRET.
138. C*****
139. CALL COMP(R,C,L,N)
140. GO TO (31,32,33) N
141. C*****
142. C** CAS OU -K-EST PLUS GRAND EN VALEUR ABSOLUE QUE-C-.ON RECOMMENCE LA DIVISION
143. C** EN REMPLACANT -A- PAR -R1-.
144. C*****
145. 32 DO 23 I=1,L
146. 23 R1(I)=R(I)
147. BOUL=.FALSE.
148. IF(R1(1).LT.0) BOUL=.TRUE.
149. GO TO 5
150. C*****
151. C** CAS OU R=C EN VALEUR ABSOLUE,IL SUFFIT D'AJOUTER OU DE RETRANCHER 1 A-Q-
152. C** SUIVANT LE SIGNE DE-R-ET ON A A=Q*C.
153. C*****
154. 31 C1=1
155. K3=L
156. IF(R(1).GT.0) GO TO 34
157. CALL DIF1(Q,C1,K3,L)
158. DO 35 I=2,L
159. 35 R(I)=0
160. R(1)=1
161. RETURN
162. 34 CALL SOM1(Q,C1,K3,L)
163. DO 36 I=2,L
164. 36 R(I)=0
165. R(1)=1
166. RETURN
167. C*****
168. C** CAS OU-R-EST PLUS PETIT QUE-C-.SI-R-EST POSITIF ON EST A LA FIN DE
169. C** L'ALGORITHME,SINON ON RETRANCHE 1 A -Q- ET -C- A -R- POUR TROUVER A=Q*C+.
170. C*****
171. 33 IF(R(1).GT.0)RETURN
172. C1=1
173. K3=L
174. CALL DIF1(Q,C1,K3,L)
175. CALL ADD(C,R,R1,L,ITE)
176. DO 37 I=1,L
177. 37 R(I)=R1(I)
178. RETURN
179. END

```

```

1.      SUBROUTINE QUOTIE(N,M,L,Q)
2.      C*****
3.      C      N ET M SONT DES TEP DE LONGUEUR L
4.      C      APPROXIMATION DU QUOTIENT Q DU RAPPORT N/M
5.      C      JM EST LE NOMBRE DE CHIFFRES AVEC LEQUEL L'ORDINATEUR TRAVAILLE
6.      C*****
7.      C***** CARTE SUSCEPTIBLE D'ETRE MODIFIEE PAR L'UTILISATEUR*****
8.      JM=15
9.      C***** FIN *****
10.     DIMENSION M(L),N(L)
11.     COMMON/BLOCK1/IBASE,LM,MK
12.     DO 1 I=2,L
13.     IF(M(I).NE.0)M1=I;GO TO 10
14.     1 CONTINUE;WRITE(MK,200);STOP
15.     DO 2 I=2,L
16.     IF(N(I).NE.0)N1=I;MA=0;IK=IBASE;GO TO 11
17.     2 CONTINUE;Q=0;RETURN
18.     11 IK=IK/10;IF(IK.GT.0)MA=MA+1;GO TO 11
19.     NA=JM/MA;IA=M(N1);NC=1
20.     13 IA=IA/10;IF(IA.LE.0)IA=N(N1);NC=1;GO TO 14
21.     MC=NC+1;GO TO 13
22.     14 IA=IA/10;IF(IA.LE.0)B=M(M1);GO TO 15
23.     NC=NC+1;GO TO 14
24.     15 Q=N(I)*M(I);IF((M1.GE.L).AND.(N1.GE.L))Q=(Q*N(L))/M(L);RETURN
25.     KA=NC;KB=MC;I1=1;I2=1;IF(M1.GE.L)I1=0;GO TO 30
26.     DO 3 J=M1+1,L
27.     IF(J-M1.GT.NA)IM=JM-NC-NA+MA;GO TO 18
28.     KB=KB+MA;IF(KB.GT.JM)IM=JM-KB+MA;GO TO 18
29.     3 B=B+IBASE+M(J);I1=0;GO TO 30
30.     18 IM1=MA-IM-1;IF(IM1.GT.0)GO TO 19
31.     IB=M(J)/10;IA=MOD(M(J),10);IF(IA.GE.5)IB=IB+1
32.     GO TO 12
33.     19 IA=M(J)/(10**IM1);IB=IA/10;IA=MOD(IA,10);IF(IA.GE.5)IB=IB+1
34.     12 B=B+10**IM+IB
35.     30 A=N(N1);IF(N1.GE.L)I2=0;GO TO 23
36.     DO 4 J=N1+1,L
37.     IF(J-N1.GT.NA)IV=JM-NA-MA-NC;GO TO 36
38.     KA=KA+MA;IF(KA.GT.JM)IV=JM-KA+MA;GO TO 36
39.     4 A=A+IBASE+N(J);I2=0;GO TO 23
40.     36 IM1=MA-IV-1;IF(IM1.GT.0)GO TO 29
41.     IB=N(J)/10;IA=MOD(N(J),10);IF(IA.GE.5)IB=IB+1
42.     GO TO 22
43.     29 IA=N(J)/(10**IM1);IB=IA/10;IA=MOD(IA,10);IF(IA.GE.5)IB=IB+1
44.     22 A=A+10**IV+IB
45.     23 J2=NC+(L-N1)-MA-JM;J1=NC+(L-M1)-MA-JM
46.     IF((I1.NE.0).AND.(I2.NE.0))GO TO 40
47.     Q=(Q+A)/B;IF(J2.GT.0)Q=Q*10.**J2;RETURN
48.     IF(J1.GT.0)Q=Q/10.**J1
49.     RETURN
50.     40 Q=(Q+A)/B;MD=NC-MC+(M1-N1)-MA;IF(MD.NE.0)Q=Q*10.**MD
51.     200 FORMAT('X','DIVISION PAR ZERO DANS QUOTIE')
52.     RETURN
53.     END

```

```

1.      FUNCTION IZERO(N,L)
2.      C*****
3.      C      TEST SUR LA NULLITE DU TEP -N- DE LONGUEUR -L-
4.      C      IZERO=0,SI LE TEP -N- EGAL ZERO
5.      C      2,SINON
6.      C*****
7.      DIMENSION N(L)
8.      DO 2 I=2,L
9.      IF(N(I).NE.0)IZERO=2;RETURN
10.     2 CONTINUE;IZERO=0;RETURN
11.     END

```



Les algorithmes de calcul sont les mêmes que pour l'arithmétique réelle simple ou double précision, nous allons donner le mode d'emploi des sous-programmes écrits en arithmétique rationnelle étendue.

Dans ces sous-programmes on note :

$$\begin{aligned}c_j &= NC(J, \cdot)/LC(J, \cdot) \\q_k^j &= NQ(J, K, \cdot)/LQ(J, K, \cdot) \\l_k^j &= NE(J, K, \cdot)/LE(J, K, \cdot) \\C_k^j &= NCR(J, K, \cdot)/LCR(J, K, \cdot) \\B_k^j &= NBR(J, K, \cdot)/LBR(J, K, \cdot)\end{aligned}$$

$$(n_i, k_i, r_i) = (NA(I), KA(I), NRA(I))$$

Le tableau NDF est défini comme dans le sous-programme QUDE. Tous ces tableaux sont échangés entre programmes principaux et les sous-programmes QUDEA, PADERA, PA2POI, SRIFON, ESTRA et DETECT par l'intermédiaire de l'instruction COMMON.

NM : plus grand entier positif représentable en machine,

IN : numéro logique de l'imprimante,

NI : " " " " " "

KN : numéro logique du lecteur de cartes,

IBASE : base dans laquelle sont écrits les T.E.P.,

LM : longueur maximale en chiffres des T.E.P. intervenant dans les programmes.

P : précision demandée dans la représentation d'un nombre sous forme rationnelle.

VIII.2. SOUS-PROGRAMMES DE CALCUL DE LA TABLE qd PAR LES METHODES NORMALE ET PROGRESSIVE

* * * QUDERA * * * *

Appel : CALL QUDERA(NL, MOI, MA, BLOC, N1, N2, N3, N4, N5, N6, N7, N8, NX, LX, NY, LY, NZ, LZ, LK, LR)

Paramètres : NL, MOI, MA et BLOC sont définis comme dans QUDE.

N1, N2, N3, N4, N5, N6, N7, N8, NX, LX, NY, LY, NZ, LZ sont des T.E.P. de longueur LK.

LK et LR sont des entiers que calcule le sous-programme INI.

Tous les paramètres sont des paramètres d'entrée, à l'exception de BLOC qui est un paramètre de sortie.

Sous-programmes utilisés : DETECT et ESTRA.

Description : La même que QUDE

* * * ESTRA * * * *

Appel : CALL ESTRA(N, NR, K, LK, MON, MAX, N1, N2, N3, N4, N5, N6, N7, N8, NX, LX, LY, NY, NZ, LZ)

Paramètres : Les paramètres N, NR, K, MON et MAX sont les mêmes que dans le sous-programme EST.

N1, N2, N3, ..., N8, NX, LX, NY, LY, NZ et LZ sont des T.E.P. de longueur LK.

Description : Même description que EST.

Remarque : Le sous-programme DETECT que l'on utilise est celui de II.

EXEMPLE D'APPEL DE L'ENSEMBLE DES SOUS-PROGRAMMES

On présente un programme qui lit les valeurs de NL, de MON, de MAX et des coefficients de la série formelle avant d'appeler les sous-programmes qui calculent la table qd.

```

1.  C*****
2.  C    PROGRAMME PRINCIPAL POUR LE CALCUL DES ELEMENTS DE LA TABLE Q-D
3.  C*****
4.  C*****CARTES POUVANT ETRE CHANGEES PAR L'UTILISATEUR*****
5.  COMMON NM,NL,P/T/NCR(-10:10,0:10,11),LCR(-10:10,0:10,11),LBR(-10:1
6.  10,0:10,11),NR(-10:10,0:10,11),LQ(-10:10,0:10,11),LE(-10:10,0:10,11
7.  1),NE(-10:10,0:10,11),NBR(-10:10,0:10,11),LC(-10:10,11),NC(-10:10,1
8.  11),NDF(-10:10,0:10)/S/NA(10),KA(10),NRA(10)/BLCK1/TRASF,LM,IN,KN
9.  DIMENSIONDT(-10:10),NX(11),X(11),NY(11),LY(11),NZ(11),LZ(11)
10. DIMENSIONN1(11),N2(11),N3(11),N4(11),N5(11),N6(11),N7(11),N8(11)
11. NM=2147483647;NI=108;P=1.D-14;IRASE=100000000;IN=108;L=80;KA=105
12. C*****FIN*****
13. READ(KN,10)NL
14. READ(KN,10)MON
15. READ(KN,10)MAX
16. READ(KN,20)(DT(I),I=MON,NL)
17. 20 FORMAT(5F15.8)
18. 10 FORMAT(I3)
19. CALL INI(LK,LR);DEPS=1./NM
20. DO 1 J=MON,NL
21. WRITE(IN,400)J,DT(J);ITA=0
22. INA=0;IDA=1;IF(ABS(DT(J)).GE.DEPS)CALL RAT(DT(J),INA,IDA,R,ITA)
23. IF(ITA.EQ.1)STOP
24. CALL STO(INA,LK,NY);CALL STO(IDA,LK,LY)
25. 1 CALL MISE(J,NY,LY,LK,NC,LC)
26. 400 FORMAT(3X,'C(',I4,')= ',D22.15)
27. CALL QUERA(NL,MON,MAX,LA,N1,N2,N3,N4,N5,N6,N7,N8,NX,LX,NY,LY,NZ,L
28. 1Z,LK,LR)
29. STOP
30. END

```

ESSAI NUMERIQUE

Pour NL = 10, MON = 0, MAX > 0 (méthodes normale et progressive) et les coefficients c_i suivants :

```

C( 0)= .0000000000000000
C( 1)= .0000000000000000
C( 2)= .1000000000000000
C( 3)= .1000000000000000
C( 4)= .1000000000000000
C( 5)= .1000000000000000
C( 6)= .1000000000000000
C( 7)= .1000000000000000
C( 8)= .0000000000000000
C( 9)= .2500000000000000
C(10)= .3000000000000000

```

les résultats sont :

Les valeurs suivies de (*) sont celles des éléments communs aux tables qd par les méthodes normale et progressive.

1) Résultats par la méthode normale

Le tableau suivant :

$N = 0$
 $N = 1$ +
 $N = 2$ + 0
 $N = 3$ + 0 0
 $N = 4$ + + + +
 $N = 5$ + + + + +
 $N = 6$ + + + + + +
 $N = 7$ + + 0 0 + +
 $N = 8$ + + 0 0 +
 $N = 9$ + + + +
 $N = 10$ + 0 +
 $N = 11$ + +



où les numéros de gauche sont les indices des diagonales et les + (resp. 0) donnent les positions des polynômes n'appartenant pas (resp. appartenant) aux blocs,

donne l'allure de la table des polynômes orthogonaux.

Les deux dernières colonnes des q et e de la table qd sont :

INDICE DE COLONNE= 4

+ 100000
-----=Q(0, 4)= .7319885900625590-05 (*)
+ 13661415131

- 136614152208209
-----=E(0, 4)= -.1369012354679980+06 (*)
+ 997902990

- 11089000
-----=Q(1, 4)= -.1369012345679010+06 (*)
+ 81

+ 11086894
-----=E(1, 4)= .1366752345679010+06 (*)
+ 81

Q(2, 4) INFINI OU INDEFINI

+ 6761
-----=E(2, 4)= .751222222222220+03 (0)
+ 9

Q(3, 4) INFINI OU INDEFINI

INDICE DE COLONNE= 5

+ 13658820583220000
-----=Q(0, 5)= .1368752336679760+06 (*)
+ 997902996661

- 626727688071553
-----=E(0, 5)= -.4588446565588540-02 (*)
+ 136588206730298414

- 60849
-----=Q(1, 5)= -.5488372126584780-02 (*)
+ 11086894

(0) valeur de l'élément appartenant à un bloc qd.

+ 0
 -----=Q(-1, 4)= .0000000000000000+00
 + 1

 + 1231855921
 -----=E(-1, 4)= .9999000965113850+00
 + 1231979000

 + 100000
 -----=Q(0, 4)= .7319885900625590-05 (*)
 + 13661415131

 - 136614152208209
 -----=E(0, 4)= -.1369012354679960+06 (*)
 + 997902990

 - 11089000
 -----=Q(1, 4)= -.1369012345679010+06 (*)
 + 81

 + 11086894
 -----=E(1, 4)= .1368752345679010+06 (*)
 + 81

 Q(2, 4) INFINI OU INDEFINI
 - 6761
 -----=E(2, 4)= -.751222222222220+03 (*) (O)
 + 9

Q(3, 4) INFINI OU INDEFINI

INDICE DE COLONNE= 5

+ 0
 -----=Q(-2, 5)= .0000000000000000+00
 + 1

 - 229639459501
 -----=E(-2, 5)= -.2301450199846790-02
 + 99780329601000

 - 123197900810000000
 -----=Q(-1, 5)= -.1002201546711240+01
 + 122927270681510379

 - 307313242891498642307971
 -----=E(-1, 5)= -.2499959847536120+02
 + 12292727148973104876810

 + 136588205832200000
 -----=Q(0, 5)= .1368752336679760+06 (*)
 + 997902996561

 - 626727688071553
 -----=E(0, 5)= -.4588446565588540-02 (*)
 + 136588206730298614

 - 60849
 -----=Q(1, 5)= -.5488372126584780-02 (*)
 + 11086894

(O) valeur de l'élément appartenant à un bloc qd.

INDICE DE COLONNE= 6

```

+ 0
-----=Q(-3, 6)= .0000000000000000+00
+ 1

- 2499960392476599401
-----=E(-3, 6)= -.1088645826770770+05
+ 229639459501000

- 24944681921647904900000000
-----=Q(-2, 6)= -.1088645596025750+05
+ 22913500958405270989101

+ 6236140587023542774043696901855041
-----=E(-2, 6)= .1088657855910290+05
+ 572828327391159091246504945490

+ 77203221340555899759689500000
-----=Q(-1, 6)= .2512219132066990+02
+ 3073132440120178779614512969

```

INDICE DE COLONNE= 7

```

+ 0
-----=Q(-4, 7)= .0000000000000000+00
+ 1

+ 294970125408609250699
-----=E(-4, 7)= .1179899194786420+00
+ 2499960392476599401000

+ 624986970073192069014410000000000
-----=Q(-3, 7)= .1088657625762720+05
+ 574089553302234113236140358901

```

Les éléments communs aux tables qd calculées par les méthodes normale et progressive qui ne sont pas dans les blocs sont égaux.




```

1.      SUBROUTINE QUADERA(IR,MOI,MA,BLOC,N1,N2,N3,N4,N5,N6,N7,N8,NX,LX,NY,
2.      11Y,NZ,LZ,LK,LR)
3.      C*****
4.      C      CALCUL DES ELEMENTS DE LA TABLE Q-D PAR LES METHODES NORMALE ET
5.      C      PROGRESSIVE
6.      C*****
7.      C*****CARTES SUSCEPTIBLES D'ETRE MODIFIEES PAR L'UTILISATEUR*****
8.      COMMON NM,N1,P/T/NCR(-10,10,0,10,11),LCR(-10,10,0,10,11),LBN(-10,1
9.      10,0,10,11),NU(-10,10,0,10,11),IQ(-10,10,0,10,11),LE(-10,10,0,10,11
10.     1),NE(-10,10,0,10,11),NBR(-10,10,0,10,11),LC(-10,10,11),NC(-10,10,1
11.     11),NDF(-10,10,0,10)/S/NA(20),KA(20),NRA(20)/BLOC1/IBASE,L4,IN,KW
12.     DIMENS UNRX(160)
13.     DIMENS UNNX(LK),LX(LK),NY(LK),LY(LK),NZ(LK),LZ(LK),NF(-20,20,0,20)
14.     C*****
15.     DIMENS UNY1(LK),N2(LK),N3(LK),N4(LK),N5(LK),N6(LK),N7(LK),N8(LK)
16.     INTEGER BLOC,BLOC1,COL
17.     NL=IR;MAX=MA;MON=MOI;M=NL;IF(MAX)2000,4000,4000
18.     C*****
19.     C      CALCUL DES COEFFICIENTS DE LA SERIE RECIPROQUE
20.     C*****
21.     2000 MAX=-1,WRITE(IN,4005)
22.     DO 2 J=0,IR
23.     CALL RANGE(J,NC,LC,LK,NX,LX);IF(IZERO(NX,LK),NE,0)NS=J;GO TO 3
24.     2 CONTINUE;WRITE(IN,206);STOP
25.     206 FORMAT(3X,'TOUS LES C(I) SONT NULS SAUF PEUT ETRE LE DERNIER')
26.     3 CALL STO(1,LK,NX);CALL RANGE(NS,NC,LC,LK,NY,LY)
27.     CALL DIVFRA(NZ,LZ,NX,NX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,ITA)
28.     IF(ITA.EQ.1)STOP
29.     CALL MACHIN(0,0,NZ,LZ,LK,NQ,LQ)
30.     I=0;CALL QUOTIE(NZ,LZ,LK,Q);WRITE(IN,400)I,Q;NL=IR-NS;MON=NL;MENL
31.     IF(NL.EQ.0)WRITE(IN,206);STOP
32.     DO 7 I=1,NL
33.     CALL STO(0,LK,NX);CALL STO(1,LK,LX)
34.     CALL MACHIN(I,0,NX,LX,LK,NQ,LQ)
35.     DO 5 J=0,I-1
36.     CALL RANGE(NP+I-J,NC,LC,LK,NY,LY);CALL REPAR(J,0,NQ,LQ,LK,N4,LZ)
37.     CALL DIVFRA(NX,LX,NZ,LZ,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,ITA)
38.     IF(ITA.EQ.1)STOP
39.     CALL REPAR(I,0,NQ,LQ,LK,NY,LY)
40.     CALL SOUFRA(NZ,LZ,NY,LY,NX,LX,N1,N2,N3,N4,N5,N6,N7,LK,ITA)
41.     IF(ITA.EQ.1)STOP
42.     5 CALL MACHIN(I,0,NZ,LZ,LK,NQ,LQ);CALL RANGE(NS,NC,LC,LK,NY,LY)
43.     CALL DIVFRA(NX,LX,NZ,LZ,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,ITA)
44.     IF(ITA.EQ.1)STOP
45.     CALL QUOTIE(NX,LX,LK,Q);WRITE(IN,400)I,Q
46.     7 CALL MACHIN(I,0,NX,LX,LK,NQ,LQ)
47.     4005 FORMAT(1X,'LES COEFFICIENTS DE LA SERIE RECIPROQUE SONT: ')
48.     400 FORMAT(3X,'C(',I2,')= ',D22,15)
49.     DO 8 I=MON,NL
50.     CALL STO(0,LK,NX);CALL STO(1,LK,LX);IF(I,LE,-1)GO TO 8
51.     CALL REPAR(I,0,NQ,LQ,LK,NX,LX)
52.     8 CALL MIDE(I,NX,LX,LK,NC,LC)
53.     4000 I=0
54.     2100 DO 9 J=MON,M+2
55.     9 NDF(J,1)=1;IF(M,LE,MON)GO TO 10
56.     M=M-2;I=I+1;GO TO 2100
57.     10 DO 11 J=MON,IR
58.     11 NF(I,0)=1;COL=1;J=MON;BLOC=0;CALL RANGE(J,NC,LC,LK,NA,LX)
59.     C*****
60.     C      DETECTION DES BLOCS APPARAISSANT A LA PREMIERE COLONNE
61.     C*****
62.     IF(IZERO(NX,LK),NE,0)GO TO 12
63.     BLOC=BLOC+1;KA(BLOC)=1;NRA(BLOC)=0;NA(BLOC)=J;J=J+1
64.     13 IF(J,GE,NL)WRITE(IN,206);STOP
65.     CALL RANGE(J,NC,LC,LK,NX,LX);IF(IZERO(NX,LK),NE,0)GO TO 12
66.     NRA(BLOC)=NRA(BLOC)+1;J=J+1;GO TO 13
67.     12 CALL RANGE(J,NC,LC,LK,NX,LX);CALL RANGE(J+1,NC,LC,LK,NY,LY)
68.     CALL DIVFRA(NZ,LZ,NY,LY,NX,LX,N1,N2,N3,N4,N5,N6,N7,N8,LK,ITA)
69.     IF(ITA.EQ.1)STOP
70.     CALL MACHIN(J,1,NZ,LZ,LK,NQ,LQ);IF(J,GE,NL-1)GO TO 13
71.     CALL RANGE(J+1,NC,LC,LK,NX,LX);IF(IZERO(NX,LK),NE,0)J=J+1;GO TO 12
72.     BLOC=BLOC+1;NRA(BLOC)=0;NA(BLOC)=J+1;KA(BLOC)=1;J=J+2
73.     14 IF(J,GE,NL)GO TO 153
74.     CALL RANGE(J,NC,LC,LK,NX,LX);IF(IZERO(NX,LK),NE,0)GO TO 12
75.     NRA(BLOC)=NRA(BLOC)+1;J=J+1;GO TO 14
76.     153 CALL RANGE(NL,NC,LC,LK,NX,LX);IF(IZERO(NX,LK),NE,0)GO TO 15
77.     NRA(BLOC)=NRA(BLOC)+1
78.     15 WRITE(IN,214);IF(BLOC,LE,0)GO TO 16
79.     214 FORMAT(3X,'LES BLOCS SONT: ')
80.     215 FORMAT(3X,'NU=',I2,3X,'NE=',I4,3X,'K=',I2,3X,'R=',I2)
81.     C*****
82.     C      DELIMITATION DES BLOCS DEMARRANT A LA PREMIERE COLONNE
83.     C*****
84.     DO 17 I=1,BLOC
85.     WRITE(IN,215)/L,NA(L),KA(L),NRA(L)
86.     DO 17 J=0,NRA(L)
87.     DO 17 I=0,NRA(L)
88.     17 NDF(NA(L)+J-I,KA(L)+I)=0

```



```

89. 16 DO 19 J=MON,NL-2
90. IF((NDF(J,1).EQ.0).OR.(NDF(J+1,1).EQ.0)) GO TO 19
91. CALL REPAR(J+1,1,NO,LO,LK,NX,LY);CALL REPAR(J,1,N2,LO,LK,NX,LX)
92. CALL MULFRA(NZ,LZ,NY,LY,NX,LX,N1,N2,N3,N4,N5,N6,N7,LK,ITA)
93. IF(ITA.EQ.1)STOP
94. 19 CALL MACHIN(J,1,NZ,LZ,LK,NE,LE);LIMITE=NL-2;COL=1
95. 24 J=MON;BLOC1=BLOC
96. C*****
97. C DETECTION DES AUTRES BLOCS
98. C*****
99. 23 IF((NDF(1,COL).EQ.0).OR.(NDF(1+1,COL).EQ.0))GO TO 20
100. CALL REPAR(1,COL,NE,LE,LK,NX,LY);IF(IZERO(NX,LK).NE.U)GO TO 20
101. BLOC=BLOC+1;NRA(BLOC)=0;KA(BLOC)=COL+1;NA(BLOC)=1;I=I+1
102. 21 IF(1.GT.LIMITE)GO TO 22
103. IF(NDF(1,COL).EQ.U)GO TO 20
104. CALL REPAR(1,COL,NE,LE,LK,NX,LX);IF(IZERO(NX,LK).NE.U)GO TO 20
105. NRA(BLOC)=NRA(BLOC)+1;I=I+1;GO TO 21
106. 20 IF(1.GT.LIMITE)GO TO 22
107. I=I+1;GO TO 23
108. 22 IF(BLOC1.GE.BLOC)GO TO 64
109. C*****
110. C DELIMITATION DES AUTRES BLOCS
111. C*****
112. DO 25 J=BLOC1+1,BLOC
113. WRITE(1,N,215)J,NA(J),KA(J),NRA(J)
114. DO 25 L=0,NRA(J)
115. DO 25 I=0,NRA(J)
116. 25 NDF(NA(J)+L-1,KA(J)+I)=0
117. 64 COL=COL+1;LIMITE=LIMITE-1;J=MON
118. C*****
119. C CALCUL DES COEFFICIENTS Q(N,J)
120. C*****
121. 27 IF((NDF(J+2,COL-2).EQ.0).AND.(NDF(J+1,COL-1).EQ.1))GO TO 28
122. IF(NDF(J,COL).EQ.0) GO TO 30
123. IF(NDF(J+1,COL).EQ.0)JL=J;CALL STO(0,LK,NZ);CALL STO(1,LK,LZ);GO TO 88
124. 10 88
125. IF((NDF(J+2,COL-1).EQ.0).OR.(NDF(J+1,COL-1).EQ.0))JL=J;GO TO 32
126. IF(NDF(J,COL-1).EQ.0)JL=J;GO TO 33
127. IF(NDF(J+1,COL-2).EQ.0)GO TO 37
128. CALL REPAR(J+1,COL-1,NO,LO,LK,NX,LX)
129. CALL REPAR(J+1,COL-1,NE,LE,LK,NY,LY)
130. CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,ITA)
131. IF(ITA.EQ.1)STOP
132. CALL REPAR(J,COL-1,NE,LE,LK,NX,LX)
133. CALL DIVFRA(NY,LY,NZ,LZ,NX,LX,N1,N2,N3,N4,N5,N6,N7,N8,LK,ITA)
134. IF(ITA.EQ.1)STOP
135. CALL MACHIN(J,COL,NY,LY,LK,NO,LO);GO TO 30
136. 33 CALL REPAR(J+1,COL-1,NO,LO,LK,NX,LX)
137. CALL REPAR(J+1,COL-1,NE,LE,LK,NY,LY)
138. CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,ITA)
139. IF(ITA.EQ.1)STOP
140. 88 CALL MACHIN(J,COL,NZ,LZ,LK,NO,LO);GO TO 40
141. 32 CALL DETECT(J,1,COL,BLOC1)
142. CALL EXTRA(NA(L),NRA(L),KA(L),LK,MON,LIMITE,N1,N2,N3,N4,N5,N6,N7,N
143. 8,NX,LX,NY,LY,NZ,LZ);J=NA(L)-1;GO TO 40
144. 28 CALL DETECT(J,2,COL,BLOC1);IF(J.GT.NA(L)-NRA(L)-2)GO TO 35
145. IF(NDF(J,COL).EQ.0)J=J+1;GO TO 35
146. IF(NDF(J+1,COL).EQ.0)GO TO 41
147. CALL REPAR(J+1,COL,NCR,LCR,LK,NX,LX);NX(1)=-NX(1)
148. CALL REPAR(J,COL-1,NE,LE,LK,NY,LY)
149. CALL DIVFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,ITA)
150. IF(ITA.EQ.1)STOP
151. 89 CALL MACHIN(J,COL,NZ,LZ,LK,NO,LO);J=J+1;GO TO 35
152. 41 CALL STO(0,LK,NZ);CALL STO(1,LK,LZ);GO TO 89
153. 35 IF(J.GT.LIMITE)GO TO 30
154. IF(J.GE.NA(L)-1)J=J-1;GO TO 30
155. IF(NDF(J,COL).EQ.0)J=J+1;GO TO 35
156. IF(NDF(J+1,COL).EQ.0)GO TO 38
157. CALL REPAR(J+1,COL,NCR,LCR,LK,NX,LX)
158. CALL REPAR(J,COL,NCR,LCR,LK,NY,LY)
159. CALL DIVFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,ITA)
160. IF(ITA.EQ.1)STOP
161. 91 CALL MACHIN(J,COL,NZ,LZ,LK,NO,LO);J=J+1;GO TO 35
162. 38 CALL STO(0,LK,NZ);CALL STO(1,LK,LZ);GO TO 91
163. 37 CALL REPAR(J+1,COL-1,NO,LO,LK,NX,LX)
164. CALL REPAR(J+1,COL-1,NE,LE,LK,NY,LY)
165. CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,ITA)
166. IF(ITA.EQ.1)STOP
167. CALL REPAR(J,COL,NCR,LCR,LK,NX,LX);NZ(1)=-NZ(1)
168. CALL DIVFRA(NY,LY,NZ,LZ,NX,LX,N1,N2,N3,N4,N5,N6,N7,N8,LK,ITA)
169. IF(ITA.EQ.1)STOP
170. 66 CALL MACHIN(J,COL,NY,LY,LK,NO,LO);GO TO 30
171. 39 CALL STO(0,LK,NY);CALL STO(1,LK,LY);GO TO 66
172. 40 IF(JL.E.MON)GO TO 30
173. IF((NDF(JL-1,COL).EQ.0).OR.(NDF(JL,COL-1).GT.0))GO TO 30
174. JL=JL-1;CALL DETECT(JL,1,COL,BLOC1);M1=NA(L);M2=NRA(L);M3=KA(L)
175. IF((M3.LT.2).OR.(NDF(M1+1,M3-2).EQ.0))GO TO 208
176. CALL REPAR(M1,M3+M2+1,NO,LO,LK,NZ,LZ);NZ(1)=-NZ(1)

```

```

177. CALL REPARE(M1+1,M3+M2,NCP,LCR,IK,NY,LY)
178. CALL ADDFRA(NX,LX,NY,LY,NZ,LZ,N1,N2,N3,N4,N5,N6,N7,LK,ITA)
179. IF(ITA.EQ.1)STOP
180. CALL MACHIN(M1,M2+M3+1,NX,LX,LK,NBR,LBR);GO TO 42
181. 208 CALL EGAL(-1,M1,M2+M3+1,NQ,LQ,IK,M1,M2+M3+1,NBR,LBR)
182. 42 IF(NDF(M1-1,M3-1),EQ.0)GO TO 64
183. CALL REPARE(M1,M2+M3+1,NBR,LBR,LK,NZ,LZ)
184. CALL REPARE(M1-1,M2+M3+1,NCR,LCR,LK,NX,LX)
185. CALL SOUFRA(NY,LY,NZ,LZ,NX,LX,N1,N2,N3,N4,N5,N6,N7,LK,ITA)
186. IF(ITA.EQ.1)STOP
187. CALL MACHIN(M1-1,M3+M2+2,NY,LY,LK,NCR,LCR);GO TO 55
188. 43 CALL EGAL(1,M1,M2+M3+1,NBR,LBR,LK,M1-1,M2+M3+2,NCR,LCR)
189. 55 CALL EGAL(1,M1-1,M3+M2+2,NCR,LCR,LK,M1-1,M3+M2+1,NE,LE)
190. 30 IF(J.GE.LIMITE)LIMITE=LIMITE-1,J=MON;GO TO 99
191. J=J+1;GO TO 27
192. 99 IF(LIMITE,LT,MON)GO TO 701
193. C*****
194. C CALCUL DES COEFFICIENTS E(N,J)
195. C*****
196. 44 IF((NDF(J+2,COL-2),EQ.0).AND.(NDF(J+1,COL-1),EQ.1))GO TO 45
197. IF((NDF(J,COL),EQ.0).OR.(NDF(J+1,COL),EQ.0))GO TO 47
198. IF((NDF(J+2,COL-1),EQ.0).OR.(NDF(J+1,COL-1),EQ.0))GO TO 48
199. IF(NDF(J,COL-1),EQ.0)GO TO 49
200. CALL REPARE(J+1,COL,NQ,LQ,LK,NZ,LZ)
201. CALL REPARE(J,COL,NQ,LQ,LK,NX,LX)
202. CALL SOUFRA(NY,LY,NZ,LZ,NX,LX,N1,N2,N3,N4,N5,N6,N7,LK,ITA)
203. IF(ITA.EQ.1)STOP
204. CALL REPARE(J+1,COL-1,NE,LE,LK,NX,LX)
205. CALL ADDFRA(NZ,LZ,NY,LY,NX,LX,N1,N2,N3,N4,N5,N6,N7,LK,ITA)
206. IF(ITA.EQ.1)STOP
207. CALL MACHIN(J,COL,NZ,LZ,LK,NE,LE);GO TO 47
208. 49 CALL REPARE(J+1,COL,NQ,LQ,LK,NZ,LZ)
209. CALL REPARE(J+1,COL-1,NE,LE,LK,NX,LX)
210. CALL ADDFRA(NY,LY,NZ,LZ,NX,LX,N1,N2,N3,N4,N5,N6,N7,LK,ITA)
211. IF(ITA.EQ.1)STOP
212. CALL MACHIN(J,COL,NY,LY,LK,NE,LE);GO TO 47
213. 48 CALL DETECT(J,1,COL,BLOC,L);J=NA(L)-1;GO TO 47
214. 45 CALL DETECT(J,2,COL,BLOC,L)
215. 50 IF((J.GT.LIMITE).OR.(J.GT.NA(L)-2))J=J-1;GO TO 47
216. IF((NDF(J,COL),EQ.0).OR.(NDF(J+1,COL),EQ.0))J=J+1;GO TO 50
217. CALL REPARE(J+1,COL,NQ,LQ,LK,NX,LX)
218. CALL REPARE(J,COL,NQ,LQ,LK,NY,LY)
219. CALL SOUFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,LK,ITA)
220. IF(ITA.EQ.1)STOP
221. CALL MACHIN(J,COL,NZ,LZ,LK,NE,LE);J=J+1;GO TO 50
222. 47 IF(J.GE.LIMITE)GO TO 51
223. J=J+1;GO TO 44
224. 51 IF(LIMITE,LE,MON)GO TO 701
225. GO TO 74
226. 701 IF(MAX,LT,0)NL=IR;LIMITE=NL-NS-1;M=1;J=1;GO TO 2004
227. MAX=MAX+1;LIMITE=NL-1;I=1;J=MON
228. C*****
229. C SORTIE DE LA TABLE Q-D CALCULEE PAR LA METHODE NORMALE
230. C*****
231. WRITE(IN,1002);WRITE(IN,2001);WRITE(IN,1003)
232. 1003 FORMAT(//)
233. 1002 FORMAT(//)
234. 866 WRITE(IN,1004);WRITE(IN,1000);WRITE(IN,1003)
235. 806 IF(NDF(J,I),EQ.0)GO TO 702
236. IF(NDF(J+1,I-1),EQ.0)GO TO 704
237. CALL REPARE(J,I,NQ,LQ,LK,NX,LX);CALL QUOTIE(NX,LX,LK,4)
238. CALL ECR(NX,LK,RX,LR);WRITE(IN,880)J,I,0;CALL ECR(LX,LK,RX,LR)
239. WRITE(IN,1004);IF(J.GE.LIMITE)GO TO 900
240. IF(NDF(J+1,I),EQ.0)GO TO 815
241. CALL REPARE(J,I,NE,LE,LK,NX,LX);CALL QUOTIE(NX,LX,LK,4)
242. CALL ECR(NX,LK,RX,LR);WRITE(IN,889)J,I,0;CALL ECR(LX,LK,RX,LR)
243. GO TO 814
244. 704 WRITE(IN,850)J,I;WRITE(IN,1002);IF(J.GE.LIMITE)GO TO 900
245. CALL REPARE(J,I,NE,LE,LK,NX,LX);CALL QUOTIE(NX,LX,LK,4)
246. CALL ECR(NX,LK,RX,LR);WRITE(IN,889)J,I,0;CALL ECR(LX,LK,RX,LR)
247. WRITE(IN,1002);GO TO 814
248. 702 WRITE(IN,850)J,I;WRITE(IN,1002);IF(J.GE.LIMITE)GO TO 900
249. 815 WRITE(IN,856)J,I;GO TO 814
250. 814 IF(J.GE.LIMITE)GO TO 900
251. J=J+1;WRITE(IN,1002);GO TO 806
252. 900 LIMITE=LIMITE-2;IF(LIMITE,LT,MON)I=0;LIMITE=NL+1;GO TO 936
253. I=I+1;I=MON;GO TO 866
254. C*****
255. C SCHEMATISATION DE LA TABLE-P- QUAND LA TABLE Q-D EST CALCULEE
256. C AVEC LA METHODE NORMALE
257. C*****
258. 936 DO 65 J=MON,LIMITE
259. IF(NDF(J,I),EQ.1)NDF(J,I)=1H+1;GO TO 65
260. NDF(J,I)=1H0
261. 65 CONTINUE;LIMITE=LIMITE-2;IF(LIMITE,LT,MON)ME=NL/2;GO TO 935
262. I=I+1;GO TO 936
263. 935 WRITE(IN,800)MON;IF(MON,LT,0)ME=0
264. DO 97 J=MON,ME;WRITE(IN,1002);KT=J+1

```

```

265.      97 WRITE(IN,800)(KT,(NF(J-I,I),I=0,J-MON))
266.      DO 67 J=1,NL-ME;IF(ME+J.GT.NL)GO TO 862
267.      WRITE(IN,1004)KT=ME+J+1
268.      67 WRITE(IN,800)(KT,(NF(ME+J-I,I),I=0,NL-ME-J+1))
269.      862 IF(MAX,4E,0)GO TO 2000
270.      RETURN
271. C*****
272. C   SYMETRIE EN VUE D'OBTENIR LA TABLE Q-D PAR LA METHODE PROGRESSIVE
273. C*****
274. 2004 IF(NDF(1-J,M+J-1),EQ,0) GO TO 2011
275.      IF(NDF(2-J,J+M-1),EQ,0) GO TO 2010
276.      NF(NS+1,M)=1;CALL EGAL(1,1-J,M+J-1,NE,LE,LK,NS+J,M,NBR,LBR)
277.      IF(NDF(1-J,M+J),EQ,0) GO TO 2013
278.      CALL EGAL(1,1-J,J+M,NQ,LQ,LK,NS+J,M,NCR,LCR);GO TO 2013
279. 2011 IF((NDF(1-J,J+M),EQ,0).OR.(NDF(2-J,J+M-1),EQ,0))GO TO 2010
280.      NF(NS+1,M)=1
281.      -- CALL EGAL(1,1-J,M+J,NQ,LQ,LK,NS+J,M,NCR,LCR);GO TO 2013
282. 2010 NF(NS+1,M)=0
283. 2013 IF(J.GE.LIMITE) GO TO 2014
284.      J=J+1;GO TO 2004
285. 2014 IF((NDF(1-J,M+J),EQ,0).OR.(NDF(1-J,M+J),EQ,0))GO TO 2017
286.      NF(NS+J+1,M)=1;CALL EGAL(1,-J,M+J,NE,LE,LK,NS+J+1,M,NBR,LBR)
287.      GO TO 2018
288. 2017 NF(NS+J+1,M)=0
289. 2018 IF(LIMITE,LE,1)LIMITE=NL-NS-1;M=1;J=1;GO TO 3500
290.      LIMITE=LIMITE-1;M=M+1;J=1;GO TO 2004
291. 3500 NF(NS+1-J,M+J-1)=1;IF(NDF(J,M),EQ,0)GO TO 3501
292.      CALL EGAL(1,J,M,NQ,LQ,LK,NS+1-J,M+J-1,NCR,LCR)
293. 3501 IF(J.GE.LIMITE)LIMITE=LIMITE-1;M=1;J=1;GO TO 3004
294.      J=J+1;GO TO 3500
295. 3004 IF(NDF(J,M),EQ,0) GO TO 3001
296.      IF(NDF(J+1,M),EQ,0) GO TO 3000
297.      NF(NS+1-J,M+J)=1;CALL EGAL(1,J,M,NE,LE,LK,NS+1-J,M+J,NBR,LBR)
298.      IF(NDF(J,M+1),EQ,0) GO TO 3002
299.      CALL EGAL(1,J,M+1,NQ,LQ,LK,NS+1-J,M+J,NCR,LCR);GO TO 3002
300. 3001 IF((NDF(J,M+1),EQ,0).OR.(NDF(J+1,M),EQ,0))GO TO 3000
301.      NF(NS+1-J,M+J)=1;CALL EGAL(1,J,M+1,NQ,LQ,LK,NS+1-J,M+J,NCR,LCK)
302.      GO TO 3002
303. 3000 NF(NS+1-J,M+J)=0
304. 3002 IF(J.GE.LIMITE) GO TO 3003
305.      J=J+1;GO TO 3004
306. 3003 IF(LIMITE,LE,1)LIMITE=NL-1;GO TO 3005
307.      LIMITE=LIMITE-1;M=M+1;J=1;GO TO 3004
308. 3005 CALL RANGE(1,NC,LC,LK,NX,LX);CALL RANGE(0,NC,LC,LK,NY,LY)
309.      CALL DIVFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,ITA)
310.      IF(ITA,EQ,1)STOP
311.      NZ(1)=NZ(1);CALL MACHIN(NS,1,NZ,LZ,LK,NBR,LBR)
312.      DO 56 J=1,NL-NS-1
313.      CALL STU(0,LK,NX);CALL STO(1,LK,LX);NF(NS-I,I+1)=1
314.      56 CALL MACHIN(NS-I,I+1,NX,LX,LK,NBR,LBR)
315. C*****
316. C   SORTIE DE LA TABLE Q-D CALCULEE PAR LA METHODE PROGRESSIVE
317. C*****
318.      WRITE(IN,5001);WRITE(IN,1003);I=1;IS=NS
319. 3012 WRITE(IN,1000)I;J=IS;WRITE(IN,1002)
320. 3011 IF(NF(I,I),EQ,0) GO TO 3007
321.      IF(NF(J+1,I-1),EQ,0) GO TO 3008
322.      CALL REPAR(J,I,NBR,LBR,LK,NX,LX);CALL QUOTIE(NX,LX,LK,0)
323.      CALL ECK(NX,LK,RX,LR);WRITE(IN,880)J,I,0;CALL ECR(LX,LK,RX,LR)
324.      WRITE(IN,1002);IF(J.GE.LIMITE)GO TO 3010
325.      IF(NF(J+1,I),EQ,0) GO TO 3009
326.      CALL REPAR(J,I,NCR,LCR,LK,NX,LX);CALL QUOTIE(NX,LX,LK,0)
327.      CALL ECR(NX,LK,RX,LR);WRITE(IN,889)J,I,0;CALL ECR(LX,LK,RX,LR)
328.      WRITE(IN,1002);GO TO 3010
329. 3007 WRITE(IN,850)J,I;WRITE(IN,1002);IF(J.GE.LIMITE)GO TO 3010
330. 3009 WRITE(IN,856)J,I;GO TO 3010
331. 3008 WRITE(IN,850)J,I;IF(J.GE.LIMITE)GO TO 3010
332.      CALL REPAR(J,I,NCR,LCR,LK,NX,LX);CALL QUOTIE(NX,LX,LK,0)
333.      CALL ECK(NX,LK,RX,LR);WRITE(IN,889)J,I,0;CALL ECR(LX,LK,RX,LR)
334.      WRITE(IN,1002)
335. 3010 IF(J.GE.LIMITE)LIMITE=LIMITE-2;GO TO 3030
336.      J=J+1;GO TO 3011
337. 3030 IF(LIMITE,LT,IS)LIMITE=NL-NS;GO TO 3026
338.      IS=IS-1;I=I+1;GO TO 3012
339. C*****
340. C   SCHEMATISATION DE LA TABLE-P- QUAND LA TABLE Q-D EST CALCULEE
341. C   AVEC LA METHODE PROGRESSIVE
342. C*****
343. 3026 DO 79 I=NS+1,NL+1
344.      DO 77 J=0,LIMITE;IF(NF(I-J,J),EQ,1)NF(I-J,J)=1H+1;GO TO 77
345.      NF(I-J,J)=1H0
346.      77 CONTINUE
347.      79 LIMITE=LIMITE-1
348. 3025 WRITE(IN,800)NS;KT=NS+1;NF(NS,0)=1H+1;WRITE(IN,1002)
349.      WRITE(IN,800)KT,NF(NS,0)
350.      DO 70 I=1,NL-NS;WRITE(IN,1002);KT=NS+I+1
351.      70 WRITE(IN,800)(KT,(NF(NS+I-J,J),J=0,NL-NS-I+1))
352.      1000 FORMAT(//,25X,'INDICE DE COLONNE=',12,/)

```



```

353. 5001 FORMAT('X, ' Q=D PAR METHODE PROGRESSIVE ')
354. 800 FORMAT('X, ' N=' , I3, 5X, 20(A1, 5X))
355. 2001 FORMAT('X, ' Q=D PAR METHODE NORMALE ')
356. 850 FORMAT('28X, ' Q(' , I3, ' , ' , I2, ' ) INFINI OU INDEFINI ')
357. 856 FORMAT('33X, ' E(' , I3, ' , ' , I2, ' ) INFINI OU INDEFINI ')
358. 880 FORMAT('-----EQ(' , I3, ' , ' , I2, ' )=' , D22, 15)
359. 889 FORMAT('-----E(' , I3, ' , ' , I2, ' )=' , D22, 15)
360. 1)
361. RETURN
362. END

```

```

1. SUBROUTINE FSTRA(N, NR, K, LK, MON, MAX, N1, N2, N3, N4, N5, N6, N7, N8, NX, LX, N
2. 1Y, LY, NZ, LZ)
3. C*****
4. C CALCUL DES TERMES PERMETTANT DE CONTOURNER LES BLOCS
5. C*****
6. C*****CARTES SUSCEPTIBLES D'ETRE MODIFIEES PAR L'UTILISATEUR *****
7. COMMON NM, NI, P/T/NCR(-10:10, 0:10, 11), LCR(-10:10, 0:10, 11), LJR(-10:1
8. 10, 0:10, 11), NR(-10:10, 0:10, 11), LR(-10:10, 0:10, 11), LE(-10:10, 0:10, 11
9. 1), NE(-10:10, 0:10, 11), NBR(-10:10, 0:10, 11), LC(-10:10, 11), NC(-10:10, 1
10. 11), NDF(-10:10, 0:10)
11. C*****FIN*****
12. DIMENSION NI(LK), N2(LK), N3(LK), N4(LK), N5(LK), N6(LK), N7(LK), N8(LK)
13. DIMENSION NX(LK), LX(LK), NY(LK), LY(LK), NZ(LK), LZ(LK)
14. C*****
15. C CALCUL DES COEFFICIENTS -C- ET -B- QUI INTERVIENNENT DANS LES RELATIONS
16. C DE RECURRENCE A TROIS TERMES
17. C*****
18. IF(K.GT.1) GO TO 120
19. CALL STO(0, LK, NX); CALL STO(1, LK, LX)
20. CALL MACHIN(N, K+NR+1, NX, LX, LK, NCR, LCR)
21. CALL RANGE(N+NR+1, NC, LC, LK, NX, LX); CALL RANGE(N-1, NC, LC, LK, NZ, LZ)
22. CALL DIVFRA(NY, LY, NX, LX, NZ, LZ, N1, N2, N3, N4, N5, N6, N7, N8, LK, ITA)
23. NY(1)=-NY(1); CALL MACHIN(N-1, K+NR+1, NY, LY, LK, NCR, LCR); J=0
24. 1 IF((N-NR-1+J.GT.MAX).OR.(J.GT.NR)) J=NR-1; GO TO 6
25. CALL EGAL(-1, N+NR+1-J, K+J, NQ, LQ, LK, N-NR-1+J, K+NR+1, NBR, LBR)
26. J=J+1; GO TO 1
27. 120 IF(NDF(N+NR+2, K-2).EQ.0) GO TO 2
28. CALL REPAR(N+NR+1, K-1, NE, LE, LK, NX, LX); NX(1)=-NX(1)
29. CALL REPAR(N+NR+1, K-1, NQ, LQ, LK, NZ, LZ)
30. CALL MULFRA(NY, LY, NX, LX, NZ, LZ, N1, N2, N3, N4, N5, N6, N7, N8, LK, ITA)
31. IF(ITA.EQ.1) STOP
32. CALL MACHIN(N+NR+1, K, NY, LY, LK, NCR, LCR)
33. IF(N-NR-1.GT.MAX) J=1; GO TO 3
34. CALL REPAR(N+NR+1, K, NQ, LQ, LK, NZ, LZ)
35. CALL SOUFRA(NY, LY, NX, LX, NZ, LZ, N1, N2, N3, N4, N5, N6, N7, LK, ITA)
36. IF(ITA.EQ.1) STOP
37. CALL MACHIN(N-NR-1, K+NR+1, NY, LY, LK, NBR, LBR); J=1; GO TO 3
38. 2 IF(N-NR-1.GT.MAX) J=1; GO TO 3
39. CALL EGAL(-1, N+NR+1, K, NQ, LQ, LK, N-NR-1, K+NR+1, NBR, LBR); J=1
40. 3 IF(NDF(N+NR+2-J, K-2).EQ.0) GO TO 4
41. CALL REPAR(N+NR+2-J, K+J-1, NCR, LCR, LK, NX, LX)
42. CALL REPAR(N+NR+1-J, K-1, NQ, LQ, LK, NZ, LZ)
43. CALL MULFRA(NY, LY, NZ, LZ, NX, LX, N1, N2, N3, N4, N5, N6, N7, N8, LK, ITA)
44. IF(ITA.EQ.1) STOP
45. CALL MACHIN(N+NR+1-J, K+J, NY, LY, LK, NCR, LCR)
46. IF((N-NR+J-1.GT.MAX).OR.(N-NR+J.GT.N)) GO TO 5
47. CALL REPAR(N+NR+2-J, K+J-1, NCR, LCR, LK, NZ, LZ)
48. CALL REPAR(N+NR+1-J, K+J, NQ, LQ, LK, NX, LX)
49. CALL SOUFRA(NY, LY, NZ, LZ, NX, LX, N1, N2, N3, N4, N5, N6, N7, LK, ITA)
50. IF(ITA.EQ.1) STOP
51. CALL MACHIN(N-NR-1+J, K+NR+1, NY, LY, LK, NBR, LBR); GO TO 5
52. 4 CALL EGAL(1, N+NR+2-J, K+J-1, NCR, LCR, LK, N+NR+1-J, K+J, NCR, LCR)
53. IF((N-NR+J-1.GT.MAX).OR.(N-NR+J.GT.N)) GO TO 5
54. CALL EGAL(-1, N+NR+1-J, K+J, NQ, LQ, LK, N-NR-1+J, K+NR+1, NBR, LBR)
55. 5 IF(J.GE.NR+1) J=NR; GO TO 6
56. J=J+1; GO TO 3
57. 6 IF((J.LT.0).OR.(N-NR-1+J.LT.MON)) GO TO 8
58. IF(NDF(N-NR-1+J, K+NR-1-J).EQ.0) GO TO 16
59. IF(NDF(N-NR+J, K+NR-2-J).EQ.0) CALL REPAR(N-NR-1+J, K+NR-J, NCR, LCR, LK
60. 1, NX, LX); GO TO 9
61. CALL REPAR(N-NR-1+J, K+NR-1-J, NE, LE, LK, NX, LX); NX(1)=-NX(1)
62. 9 CALL REPAR(N-NR+J, K+NR+1, NCR, LCR, LK, NZ, LZ); NZ(1)=-NZ(1)
63. CALL DIVFRA(NY, LY, NZ, LZ, NX, LX, N1, N2, N3, N4, N5, N6, N7, N8, LK, ITA)

```



```

64.      IF(ITA.EQ.1)STOP
65.      CALL MACHIN(N-NR-1+J,K+NR+1,NY,LY,LK,NCR,LCR);J=J-1;GO TO 6
66.      16 CALL EGAL(1,N-NR+J,K+NR+1,NCR,LCR,LK,N-NR-1+J,K+NR+1,NCR,LCR)
67.      J=J-1;GO TO 6
68.      8 IF(N-NR-2.LT.MUN)J=0;GO TO 13
69.      C*****
70.      C      CALCUL DE Q ET E
71.      C*****
72.      IF(NDF(N-NR-2,K+NR).EQ.0)GO TO 11
73.      IF(NDF(N-NR-1,K+NR-1).EQ.0)GO TO 22
74.      CALL REPAR(N-NR-2,K+NR,NE,LE,LK,NX,LX);NX(1)=-NX(1);GO TO 12
75.      22 CALL REPAR(N-NR-2,K+NR+1,NCP,LCR,LK,NX,LX)
76.      12 CALL REPAR(N-NR-1,K+NR+1,NCR,LCR,LK,NZ,LZ)
77.      CALL DIVFRA(NY,LY,NZ,LZ,NX,LX,N1,N2,N3,N4,N5,N6,N7,N8,LK,ITA)
78.      IF(ITA.EQ.1)STOP
79.      CALL MACHIN(N-NR-2,K+NR+1,NY,LY,LK,NQ,LU)
80.      IF(N-NR-1.GT.MAX)RETURN
81.      CALL REPAR(N-NR-1,K+NR+1,NBR,LBR,LK,NX,LX);NX(1)=-NX(1)
82.      CALL SOUFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,LK,ITA)
83.      IF(ITA.EQ.1)STOP
84.      CALL MACHIN(N-NR-2,K+NR+1,NZ,LZ,LK,NE,LE);J=0;GO TO 13
85.      11 CALL EGAL(-1,N-NR-1,K+NR+1,NCR,LCP,LK,N-NR-2,K+NR+1,NQ,LU)
86.      IF(N-NR-1.GT.MAX)RETURN
87.      CALL EGAL(-1,N-NR-1,K+NR+1,NBR,LBR,LK,N-NR-2,K+NR+1,NE,LE);J=0
88.      13 IF(N-NR-1.LT.MUN)J=MUN-N+NR+1
89.      C*****
90.      C      CALCUL DES AUTRES TERMES INTERVENANT DANS LES RELATION DE RECURRENCE
91.      C      A TROIS TERMES
92.      C*****
93.      15 IF((N-NR+J.GT.MAX).OR.(J.GE.NR))RETURN
94.      IF(NDF(N-NR-1+J,K+NR-1-J).NE.0)GO TO 20
95.      CALL EGAL(1,N-NR+J,K+NR+1,NBR,LBR,LK,N-NR-1+J,K+NR+2,NCR,LCR);GO
96.      TO 14
97.      20 CALL REPAR(N-NR+J,K+NR+1,NBR,LBR,LK,NX,LX)
98.      CALL REPAR(N-NR-1+J,K+NR+1,NCP,LCR,LK,NZ,LZ)
99.      CALL SOUFRA(NY,LY,NX,LX,NZ,LZ,N1,N2,N3,N4,N5,N6,N7,LK,ITA)
100.     IF(ITA.EQ.1)STOP
101.     CALL MACHIN(N-NR-1+J,K+NR+2,NY,LY,LK,NCR,LCR)
102.     14 CALL EGAL(1,N-NR-1+J,K+NR+2,NCR,LCR,LK,N-NR-1+J,K+NR+1,NE,LE)
103.     J=J+1;GO TO 15
104.     END

```

VIII.3. SOUS-PROGRAMMES DE CALCUL DES POLYNOMES ORTHOGONAUX REGULIERS ET LEURS ASSOCIES, ET DES APPROXIMANTS DE PADE

On note :

- NP(J, I, •) et LP(J, I, •) (resp. NPR(J, I, •) et LPR(J, I, •)),
pour $0 \leq I \leq K$, les numérateurs et dénominateurs des coefficients
de x^I dans le polynôme $P_k^{(j)}(x)$ (resp. $P_{pr(k,j)}^{(j)}(x)$).
- NH(J, I, •) et LH(J, I, •) (resp. NHR(J, I, •) et LHR(J, I, •)),
pour $0 \leq I \leq K$, les numérateurs et dénominateurs des coefficients
du polynôme $Q_k^{(j)}(x)$ (resp. $Q_{pr(k,j)}^{(j)}(x)$).

* * * PADERA * * * *

Appel : CALL PADERA(NL, N1, N2, N3, ..., N7, N8, NX, LX, LY, NY, LZ, LZ,
NG, LG, LK, LR1, RZ)

Paramètres : NL est défini comme dans APADE. LK et LR1 sont les entiers calculés
par INI.

N1(•), N2(•), ..., N7(•), N8(•), NX(•), LX(•), NY(•), NZ(•), LZ(•),
NG(•), LG(•) sont des T.E.P. de longueur LK.

RZ est un tableau de longueur LR1.

Tous les paramètres sont des paramètres d'entrée.

Sous-programmes utilisés : QUDEA, DETECT, ESTRA et ECRAT si l'on veut calculer
les approximants de Padé.

Description : Même description que APADE.

* * * ECRAT * * * *

Appel : CALL ECRAT(NS, LS, NR, LR, IA, J, N1, N2, ..., N7, N8, NX, LX, NY, LY,
NZ, LZ, LK, LR1, RZ)

Paramètres : Les $NS(I, \cdot)$ et $LS(I, \cdot)$ (resp. $NR(I, \cdot)$ et $LR(I, \cdot)$) pour $0 \leq I \leq J$, sont des T.E.P. de longueur LK, ils représentent les numérateurs et dénominateurs des coefficients de x^I dans le polynôme $P_J^{(IA-J)}(x)$ (resp. $Q_J^{(IA-J)}(x)$).

Les autres paramètres ont la même définition que dans PADERA.

Tous les paramètres sont des paramètres d'entrée.

Description : Même description que ECRIRE.

EXEMPLE D'APPEL DE L'ENSEMBLE DES SOUS-PROGRAMMES

On présente un programme qui lit la valeur de NL et les coefficients de la série formelle, et fait appel à l'ensemble des sous-programmes.

```

1. C*****
2. C   PROGRAMME PRINCIPAL POUR LE CALCUL DES POLYNOMES ORTHOGONAUX REGULIERS
3. C   ET LEURS ASSOCIES, ET DONC LES APPROXIMANTS DE PADE
4. C   LES CALCULS SONT FAITS AVEC L'ARITHMETIQUE RATIONNELLE ETENDUE
5. C*****
6. C*****CARTES POUVANT ETRE CHANGEES PAR L'UTILISATEUR*****
7. COMMON NM,NI,P/T/NCR(-10:10,0:10,11),LCR(-10:10,0:10,11),LBR(-10:1
8. 10,0:10,11),NR(-10:10,0:10,11),LR(-10:10,0:10,11),LE(-10:10,0:10,1
9. 1),NE(-10:10,0:10,11),NBR(-10:10,0:10,11),LC(-10:10,1),NC(-10:10,1
10. 11),NDF(-10:10,0:10)/S/NA(10),KA(10),NRA(10)/BLCK1/TBASE,LM,IN,KN
11. DIMENSIONDT(-10:10),NX(11),LX(11),NY(11),LY(11),NZ(11),LZ(11)
12. DIMENSIONN1(11),N2(11),N3(11),N4(11),N5(11),N6(11),N7(11),N8(11)
13. DIMENSIONHG(11),LG(11),RX(160)
14. NM=2147483647;NI=108;P=1,D=14;IBASE=100000000;IN=108;LM=80;KN=105
15. C*****FIN*****
16. READ(KN,10)NL
17. READ(KN,20)(DT(I),I=0,NL)
18. 20 FORMAT(5F15,8)
19. 10 FORMAT(I3)
20. CALL INI(LK,LR);DEPS=1./NM
21. DO 1 J=0,NL
22. WRITE(IN,400)J,DT(J);ITA=0
23. INA=0;IDA=1;IF(ABS(DT(J)).GE.DEPS)CALL RAT(DT(J),INA,IDA,R,ITA)
24. IF(ITA.EQ.1)STOP
25. CALL STO(INA,LK,NY);CALL STO(IDA,LK,LY)
26. 1 CALL MISE(J,NY,LY,LK,NC,LC)
27. 400 FORMAT(3X,'C(',I4,')=' ,D22,15)
28. CALL PADERA(NL,N1,N2,N3,N4,N5,N6,N7,N8,NX,LX,NY,LY,NZ,LZ,NG,LG,LK,
29. 1LR,RX);STOP
30. END

```

ESSAI NUMERIQUE

Pour $NL = 10$ et les coefficients c_i suivants :

```

C( 0)= .1000250000000000+06
C( 1)= .6000000000000000-05
C( 2)= .0000000000000000+00
C( 3)= .0000000000000000+00
C( 4)= .0000000000000000+00
C( 5)= .1100000000000000+02
C( 6)= .9000000000000000-04
C( 7)= .9000000000000000-04
C( 8)= .9000000000000000-04
C( 9)= .9000000000000000-04
C(10)= .1000000000000000+06

```

Les résultats sont :

Le tableau

N = 0						
N = 1	+					
N = 2	+	+				
N = 3	+	+	+			
N = 4	+	0 0 0				
N = 5	+	0 0 0			+	
N = 6	+	0 0 0			+	
N = 7	+	+	+	+	+	
N = 8	+	+	+	+	+	
N = 9	+	+	0 0			
N = 10	+	+	0			
N = 11	+	+				

où les numéros de gauche sont les indices des diagonales, les + donnent les positions des polynômes orthogonaux réguliers et les 0 celles des polynômes non orthogonaux réguliers,

donne l'allure de la table P des polynômes orthogonaux.



Les polynômes $P_5^{(0)}(x)$ et $Q_5^{(0)}(x)$ sont :

Le polynôme P_k donné sous la forme : $\frac{N_0}{M_0} = \alpha_0 X^{**0}$

$$\frac{N_1}{M_1} = \alpha_1 X^{**1}$$

⋮

$$\frac{N_k}{M_k} = \alpha_k X^{**k}$$

est égal à $\sum_{s=0}^k \frac{N_s}{M_s} x^s$ (coefficients rationnels)

ou à $\sum_{s=0}^k \alpha_s x^s$ (coefficients réels)

$$P_5^{(0)}(x)$$

```
- 40262749909820319410413914590415
-----= -.1099725068727910-03 x** 0
+ 366116506250000000000098003396206561

- 2995425162404580775301900000000
-----= -.8181617357505790-05 x** 1
+ 366116506250000000000098008396206561

- 2995449670628958648600000000000
-----= -.8181604298559150-05 x** 2
+ 366116506250000000000098008396206561

- 2995474179054060000000000000000
-----= -.8181701240160200-05 x** 3
+ 366116506250000000000098008396206561

- 29954767260000000098008396206561
-----= -.8181708196814430-05 x** 4
+ 366116506250000000000098008396206561

+ 1
-----= .1000000000000000+01 x** 5
+ 1
```

$$Q_5^{(0)}(x)$$

```
- 299617401887470865073354901041600000
-----= -.8183662762336080+00 x** 0
+ 366116506250000000000098008396206561

- 29961985332263443900539360000000000
-----= -.8183729720124690+00 x** 1
+ 366116506250000000000098008396206561

- 149811152389276059280000000049020188619683
-----= -.8183796678461170+00 x** 2
+ 183058251125000000000049004198103280500000

- 149810161409506254901644914906000823880317
-----= -.8183743636363620+00 x** 3
+ 183058251125000000000049004198103280500000

+ 100025
-----= .1000250000000000+06 x** 4
+ 1
```

L'approximant de Padé $[4/5]_f(x)$ est :

APPROXIMANT DE PADE(4, 5)

NUMERATEUR

```

+ 100025
----- = .1000250000000000+06 x** 0
+ 1

- 149810181409556254901644914986600823880317
----- = -.81837436363620+00 x** 1
+ 183058253125000000000049004198103280500000

- 149811152388927605928000000294025188619683
----- = -.8183796678461170+00 x** 2
+ 183058253125000000000049004198103280500000

- 299619853322654453900539560000000000
----- = -.3183729720124690+00 x** 3
+ 3661165062500000000000098008396206561

- 299617401887470885073354901041600000
----- = -.8183662762336080+00 x** 4
+ 3661165062500000000000098008396206561

```

DENOMINATEUR

```

+ 1
----- = .1000000000000000+01 x** 0
+ 1

- 29954767260000000098008396206561
----- = -.8181758196814430-05 x** 1
+ 3661165062500000000000098008396206561

- 29954741790540600000000000000000
----- = -.8181751240160200-05 x** 2
+ 3661165062500000000000098008396206561

- 29954496706289586486000000000000
----- = -.8181684298559150-05 x** 3
+ 3661165062500000000000098008396206561

- 29954251624043807753019660000000
----- = -.8181617357505790-05 x** 4
+ 3661165062500000000000098008396206561

- 40262749999820319410413914598415
----- = -.1099725068727910-03 x** 5
+ 3661165062500000000000098008396206561

```




```

1. SUBROUTINE PAVERA(NL,N1,N2,N3,N4,N5,N6,N7,N8,NX,LX,NT,LY,NZ,LZ,NG,
2. 1LG,LK,LR1,RZ)
3. C*****
4. C SOUS PROGRAMME DE CALCUL DES POLYNOMES ORTHOGONAUX REGULIERS ET
5. C ET LEURS ASSOCIES,
6. C CE SOUS PROGRAMME UTILISE LE SOUS PROGRAMME ECRAT POUR LE CALCUL DES
7. C APPROXIMANTS DE PADE
8. C IL APPELLE QUADERA POUR LE CALCUL DE LA TABLE Q-D
9. C*****
10. C***** CARTES SUSCEPTIBLES D'ETRE MODIFIEES PAR L'UTILISATEUR*****
11. COMMON NM,N1,P/T/NCR(-10,10,0,10,11),LCR(-10,10,0,10,11),LBR(-10,1
12. 10,0,10,11),NCR(-10,10,0,10,11),LCR(-10,10,0,10,11),LE(-10,10,0,10,11
13. 1),NE(-10,10,0,10,11),NBR(-10,10,0,10,11),LCR(-10,10,0,10,11),NC(-10,10,1
14. 11),NDF(-10,10,0,10,11)/S/NA(20),KA(20),NRA(20)/BLOK1/IBASE,LW,IN,KN
15. DIMENS1UNNX(LK),LX(LK),NY(LK),LY(LK),NZ(LK),LZ(LK),NG(LK),LG(LK)
16. DIMENS1UNN1(LK),N2(LK),N3(LK),N4(LK),N5(LK),N6(LK),N7(LK),N8(LK)
17. DIMENS1UNNR(-10,10,0,10,11),NT(-10,10,0,10,11),LR(-10,10,0,10,11)
18. DIMENS1UNNW(-10,10,0,10,11),LT(-10,10,0,10,11),LW(-10,10,0,10,11)
19. DIMENS1UNNH(-10,10,0,10,11),LH(-10,10,0,10,11),NP(-10,10,0,10,11),
20. 1LP(-10,10,0,10,11),NHR(-10,10,0,10,11),LHR(-10,10,0,10,11),NPR(-10
21. 1,10,0,10,11),LPR(-10,10,0,10,11),RZ(LR1)
22. C***** FIN*****
23. C APPEL DU SOUS PROGRAMME QUADERA
24. CALL QUADERA(L,0,0,LA,N1,N2,N3,N4,N5,N6,N7,N8,NX,LX,NY,LY,NZ,LZ,LK
25. 1,LR1)
26. C*****
27. C
28. C CALCUL DES POLYNOMES ORTHOGONAUX REGULIERS DE LA DEMI-TABLE SUPERIEURE
29. C ET LEURS ASSOCIES
30. C
31. C*****
32. IA=0;ML=(NL+1)/2;CALL STO(0,LK,NX);CALL STO(1,LK,LX)
33. CALL MACHIN(IA,0,NX,LX,LK,NPR,1,PR);CALL MACHIN(IA,0,NX,LX,LK,NH,LH
34. 1);CALL MACHIN(IA,0,LX,LX,LK,NP,LP)
35. CALL MACHIN(IA,0,LX,LX,LK,NHR,LHR);IA=1
36. 100 IF(IA.GT.ML)IA=IA-1;GO TO 390
37. CALL STO(0,LK,NX);CALL STO(1,LK,LX)
38. CALL MACHIN(IA,0,NX,LX,LK,NPR,1,PR);CALL MACHIN(IA,0,NX,LX,LK,NH,LH
39. 1);CALL MACHIN(IA,0,LX,LX,LK,NP,LP)
40. CALL MACHIN(IA,0,LX,LX,LK,NHR,LHR);J=1
41. 340 JA=IA-J;IF(NDF(IA-J,J).EQ.0)WRITE(IN,350)JA,J;GO TO 101
42. CALL STO(0,LK,NX);CALL STO(1,LK,LX)
43. CALL MISE(J,LX,LX,LK,NT,LT);CALL MISE(J,NX,LX,LK,NR,LR)
44. IF(NDF(IA-J+1,J-1).EQ.0)GO TO 102
45. CALL REPAR(IA-J,J,NQ,LQ,LK,NX,LX);CALL REPAR(IA-J,0,NP,LP,LK,NY,LY
46. 1);CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
47. IF(IT.EQ.1)STOP
48. NZ(1)=NZ(1);CALL MISE(0,NZ,LZ,LK,NT,LT)
49. DO 9 I=0,J-1
50. CALL REPAR(IA-J+1,I,NP,LP,LK,NX,LX);CALL RANGE(IA-J,NG,LG,LK,NY,LY
51. 1);CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
52. IF(IT.EQ.1)STOP
53. CALL REPAR(IA-J,I,NH,LH,LK,NX,LX);CALL REPAR(IA-J,J,NQ,LQ,LK,NY,LY
54. 1);CALL MULFRA(NG,LG,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
55. IF(IT.EQ.1)STOP
56. CALL MULFRA(NX,LX,NZ,LZ,NG,LG,N1,N2,N3,N4,N5,N6,N7,LK,IT)
57. IF(IT.EQ.1)STOP
58. CALL REPAR(IA-J+1,I,NH,LH,LK,NZ,LZ)
59. CALL MULFRA(NY,LY,NX,LX,NZ,LZ,N1,N2,N3,N4,N5,N6,N7,LK,IT)
60. IF(IT.EQ.1)STOP
61. 9 CALL MISE(I,NY,LY,LK,NR,LR);IF(J.EQ.1)GO TO 120
62. 350 FORMAT(3X,'LE POLYNOME',2X,I2,1X,I2,3X,'NON O.R.')
63. DO 2 I=1,J-1
64. CALL REPAR(IA-J,I,NP,LP,LK,NX,LX);CALL REPAR(IA-J,J,NQ,LQ,LK,NY,LY
65. 1);CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
66. IF(IT.EQ.1)STOP
67. CALL REPAR(IA-J+1,I-1,NP,LP,LK,NX,LX)
68. CALL MULFRA(NY,LY,NX,LX,NZ,LZ,N1,N2,N3,N4,N5,N6,N7,LK,IT)
69. IF(IT.EQ.1)STOP
70. 2 CALL MISE(I,NY,LY,LK,NT,LT);GO TO 120
71. C*****
72. C CAS OU LE POLYNOME EST SUR L'EXT D'UN BLOC
73. C*****
74. 102 CALL DEFECT(IA-J,1,J,LA,L);IA1=NRA(L);IA2=NA(L)
75. IS=IA-J-IA2+IA1+1;IF(IA-J.GT.0)GO TO 103
76. DO 0 I=0,IS
77. CALL REPAR(IA2-IA1-1+I,J,NBR,LBR,LK,NX,LX)
78. 6 CALL MISE(IS-I,NX,LX,LK,NW,LW);CALL STO(1,LK,LX)
79. CALL MISE(IS+1,LX,LX,LK,NW,LW)
80. DO 30 I=0,J-1
81. MS=MIN(IS+1,I)
82. CALL STO(0,LK,NX);CALL STO(1,LK,LX);M4=M4X(0,I+1+IS-J)
83. CALL MISE(I,NX,LX,LK,NT,LT);CALL MISE(I,NX,LX,LK,NR,LR)
84. DO 30 V=M4,M5
85. CALL REPAR(IA-J,I-K,NP,LP,LK,NX,LX);CALL RANGE(K,NW,LW,LK,NY,LY)
86. CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
87. IF(IT.EQ.1)STOP
88. CALL RANGE(I,NT,LT,LK,NX,LX)

```



```

89. CALL ADDFRA(NY,LY,NZ,LZ,NX,LX,N1,N2,N3,N4,N5,N6,N7,LK,IT)
90. IF(IT,EQ,1)STOP
91. CALL MISE(I,NY,LY,LK,NT,LT)
92. CALL REPAR(IA-J,I-K,NH,LH,LK,NX,LX);CALL RANGE(K,NW,LW,LK,NY,LY)
93. CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
94. IF(IT,EQ,1)STOP
95. CALL RANGE(I,NR,LK,LK,NX,LX)
96. CALL ADDFRA(NY,LY,NZ,LZ,NX,LX,N1,N2,N3,N4,N5,N6,N7,LK,IT)
97. IF(IT,EQ,1)STOP
98. 30 CALL MISE(I,NY,LY,LK,NR,LR)
99. DO 16 I=0,J-IS-2
100. CALL REPAR(IA-J,J,NCR,LCR,LK,NX,LX);CALL REPAR(IA-J,I,NPR,LPR,LK,N
101. 1Y,LY);CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
102. IF(IT,EQ,1)STOP
103. CALL RANGE(I,NT,LT,LK,NX,LX)
104. CALL ADDFRA(NY,LY,NZ,LZ,NX,LX,N1,N2,N3,N4,N5,N6,N7,LK,IT)
105. IF(IT,EQ,1)STOP
106. CALL MISE(I,NY,LY,LK,NT,LT)
107. CALL REPAR(IA-J,J,NCR,LCR,LK,NX,LX);CALL REPAR(IA-J,I,NHR,LHR,LK,N
108. 1Y,LY);CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
109. IF(IT,EQ,1)STOP
110. CALL RANGE(I,NR,LK,LK,NX,LX)
111. CALL ADDFRA(NY,LY,NZ,LZ,NX,LX,N1,N2,N3,N4,N5,N6,N7,LK,IT)
112. IF(IT,EQ,1)STOP
113. 10 CALL MISE(I,NY,LY,LK,NR,LR);GO TO 120
114. 103 CALL REPAR(IA-J-1,J,NE,LE,LK,NX,LX);IF(IA-J,LE,IA2-IA1-1)NX(1)=-NX
115. 1(1)
116. DO 4 I=0,J-1
117. CALL REPAR(IA-J,I,NP,LP,LK,NY,LY)
118. CALL MULFRA(NZ,LZ,NY,LY,NX,LX,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
119. IF(IT,EQ,1)STOP
120. CALL REPAR(IA-J-1,I,NP,LP,LK,NY,LY);CALL ADDFRA(NG,LG,NZ,LZ,NY,LY,
121. 1N1,N2,N3,N4,N5,N6,N7,LK,IT);IF(IT,EQ,1)STOP
122. 4 CALL MISE(I,NG,LG,LK,NT,LT);CALL REPAR(IA-J,0,NH,LH,LK,NY,LY)
123. CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
124. IF(IT,EQ,1)STOP
125. CALL REPAR(IA-J-1,0,NP,LP,LK,NX,LX);CALL RANGE(IA-J-1,NC,LC,LK,NG,
126. 1LG);CALL MULFRA(NY,LY,NX,LX,NG,LG,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
127. IF(IT,EQ,1)STOP
128. CALL SOUFRA(NX,LX,NZ,LZ,NY,LY,N1,N2,N3,N4,N5,N6,N7,LK,IT)
129. IF(IT,EQ,1)STOP
130. CALL MISE(0,NX,LX,LK,NR,LR)
131. DO 5 I=1,J-1
132. CALL REPAR(IA-J,I,NH,LH,LK,NY,LY);CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1
133. 1,N2,N3,N4,N5,N6,N7,N8,LK,IT);IF(IT,EQ,1)STOP
134. CALL MISE(I,NZ,LZ,LK,NR,LR)
135. CALL REPAR(IA-J-1,I,NP,LP,LK,NY,LY);CALL RANGE(IA-J-1,NC,LC,LK,NG,
136. 1LG);CALL MULFRA(NZ,LZ,NY,LY,NG,LG,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
137. IF(IT,EQ,1)STOP
138. CALL RANGE(I,NR,LK,LK,NG,LG);CALL SOUFRA(NY,LY,NG,LG,NZ,LZ,N1,N2,N
139. 13,N4,N5,N6,N7,LK,IT);IF(IT,EQ,1)STOP
140. CALL REPAR(IA-J-1,I-1,NH,LH,LK,NG,1G);CALL ADDFRA(NZ,LZ,NY,LY,NG,L
141. 1G,N1,N2,N3,N4,N5,N6,N7,LK,IT);IF(IT,EQ,1)STOP
142. 5 CALL MISE(I,NZ,LZ,LK,NR,LR)
143. 120 WRITE(IN,300)JA,J
144. DO 8 I=0,J
145. CALL EGAL(1,IA-J,I,NH,LH,LK,IA-J,I,NHR,LHR)
146. CALL EGAL(1,IA-J,I,NP,LP,LK,IA-J,I,NPR,LPR)
147. CALL RANGE(I,NR,LR,LK,NX,LX);CALL MACHIN(IA-J,I,NX,LX,LK,NH,LH)
148. CALL RANGE(I,NT,LT,LK,NY,LY);CALL MACHIN(IA-J,I,NY,LY,LK,NP,LP)
149. 8 CONTINUE
150. DO 98 I=0,J
151. CALL RANGE(I,NT,LT,LK,NX,LX);CALL QUOTIE(NX,LX,LK,Q1)
152. CALL ECR(NX,LK,RZ,LR1);WRITE(IN,320)Q1,I;CALL ECR(LX,LK,RZ,LR1)
153. 98 WRITE(IN,1000);WRITE(IN,625)JA,J
154. DO 99 I=0,J-1
155. CALL RANGE(I,NR,LK,LK,NX,LX);CALL QUOTIE(NX,LX,LK,Q1)
156. CALL ECR(NX,LK,RZ,LR1);WRITE(IN,320)Q1,I;CALL ECR(LY,LK,RZ,LR1)
157. 99 WRITE(IN,1000)
158. C APPEL A ECRAT
159. CALL ECRAT(NT,LT,NR,LR,IA,J,N1,N2,N3,N4,N5,N6,N7,N8,NX,LX,NY,LY,NZ
160. 1,LZ,LK,LR1,RZ);GO TO 301
161. 101 CALL STO(0,LK,NX);CALL STO(1,LK,LX)
162. CALL MACHIN(IA-J,J,NX,LX,LK,NP,LP);CALL MACHIN(IA-J,J,NX,LX,LK,NH,
163. 1LH);CALL MACHIN(IA-J,J,NX,LX,LK,NPR,LPR)
164. CALL MACHIN(IA-J,J,NX,LX,LK,NHR,LHR)
165. 301 IF(J,LT,IA)J=J+1;GO TO 340
166. IA=IA+1;GO TO 100
167. 300 FORMAT(//,3X,'POLYNOME('',I3,'',I2,'')',/)
168. 625 FORMAT(//,3X,'ASSOCIE('',I3,'',I2,'')',/)
169. 320 FORMAT('-----',I22,15,'X+',I2)
170. 1000 FORMAT(//)
171. C*****
172. C
173. C CALCUL DES POLYNOMES ORTHOGONAUX REGULIERS DE LA DEMI-TABLE INFERIEURE
174. C ET LEURS ASSOCIES
175. C
176. C*****

```

```

177. 370 IF(MOD(NL,2).NE.0)ML=ML-1
178. 360 IA=IA+1;IF(IA.GT.NL)RETURN
179. CALL STU(0,LK,NX);CALL STO(1,LK,LX)
180. CALL MACHIN(IA,0,NX,LX,LK,NPR,LPR);CALL MACHIN(IA,0,NX,LX,LK,NH,LN
181. 1);CALL MACHIN(IA,0,LX,LX,LK,NP,LP)
182. CALL MACHIN(IA,0,LX,LX,LK,NHR,IHP);J=1
183. 370 JA=IA-J;IF(NDF(IA-J,J).EQ.0)WRITE(IN,350)JA,J;GO TO 401
184. CALL STU(0,LK,NX);CALL STO(1,LV,LX)
185. CALL MISE(J,LX,LX,LK,NT,LT);CALL MISE(J,NX,LX,LK,NR,LR)
186. IF(NDF(IA-J+1,J-1).EQ.0)GO TO 202
187. CALL REPAR(IA-J,J,NQ,LQ,LK,NX,LX);CALL REPAR(IA-J,0,NP,LP,LK,NY,LY
188. 1);CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
189. IF(IT.EQ.1)STOP
190. NZ(1)=NZ(1);CALL MISE(0,NZ,LZ,LK,NT,LT)
191. DO 29 I=0,J-1
192. CALL REPAR(IA-J+1,I,NP,LP,LK,NX,LX);CALL RANGE(IA-J,NC,LC,LK,NY,LY
193. 1);CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
194. IF(IT.EQ.1)STOP
195. CALL REPAR(IA-J,I,NH,LH,LK,NX,LX);CALL REPAR(IA-J,J,NQ,LQ,LK,NY,LY
196. 1);CALL MULFRA(NG,LG,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
197. IF(IT.EQ.1)STOP
198. CALL SOUFRA(NX,LX,NZ,LZ,NG,LG,N1,N2,N3,N4,N5,N6,N7,LK,IT)
199. IF(IT.EQ.1)STOP
200. CALL REPAR(IA-J+1,I,NH,LH,LK,NZ,LZ)
201. CALL ADDFRA(NY,LY,NZ,LZ,NX,LX,N1,N2,N3,N4,N5,N6,N7,LK,IT)
202. IF(IT.EQ.1)STOP
203. 29 CALL MISE(I,NY,LY,LK,NR,LR);IF(J.EQ.1)GO TO 220
204. DO 22 I=1,J-1
205. CALL REPAR(IA-J,I,NP,LP,LK,NX,LX);CALL REPAR(IA-J,J,NQ,LQ,LK,NY,LY
206. 1);CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
207. IF(IT.EQ.1)STOP
208. CALL REPAR(IA-J+1,I-1,NP,LP,LK,NX,LX)
209. CALL SOUFRA(NY,LY,NX,LX,NZ,LZ,N1,N2,N3,N4,N5,N6,N7,LK,IT)
210. IF(IT.EQ.1)STOP
211. 22 CALL MISE(I,NY,LY,LK,NT,LT);GO TO 220
212. C*****
213. C CAS OU LE POLYNOME EST SUR L'EST D'UN BLOC
214. C*****
215. 202 CALL DETECT(IA-J,1,J,LA,L);IA1=NRA(L);IA2=NA(L)
216. CALL REPAR(IA-J-1,J,NE,LE,LK,NX,LX);IF(IA-J,LE,IA2-IA1-1)NX(1)=-NX
217. 1(1)
218. DO 24 I=0,J-1
219. CALL REPAR(IA-J,I,NP,LP,LK,NY,LY)
220. CALL MULFRA(NZ,LZ,NY,LY,NX,LX,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
221. IF(IT.EQ.1)STOP
222. CALL REPAR(IA-J-1,I,NP,LP,LK,NY,LY);CALL ADDFRA(NG,LG,NZ,LZ,NY,LY,
223. 1,N1,N2,N3,N4,N5,N6,N7,LK,IT);IF(IT.EQ.1)STOP
224. 24 CALL MISE(I,NG,LG,LK,NT,LT);CALL REPAR(IA-J,0,NH,LH,LK,NY,LY)
225. CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
226. IF(IT.EQ.1)STOP
227. CALL REPAR(IA-J-1,0,NP,LP,LK,NX,LX);CALL RANGE(IA-J-1,NC,LC,LK,NG,
228. 1,LG);CALL MULFRA(NY,LY,NX,LX,NG,LG,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
229. IF(IT.EQ.1)STOP
230. CALL SOUFRA(NX,LX,NZ,LZ,NY,LY,N1,N2,N3,N4,N5,N6,N7,LK,IT)
231. IF(IT.EQ.1)STOP
232. CALL MISE(0,NX,LX,LK,NR,LR);IF(J.EQ.1)GO TO 220
233. DO 25 I=1,J-1
234. CALL REPAR(IA-J,I,NH,LH,LK,NY,LY);CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1
235. 1,N2,N3,N4,N5,N6,N7,N8,LK,IT);IF(IT.EQ.1)STOP
236. CALL MISE(I,NZ,LZ,LK,NR,LR)
237. CALL REPAR(IA-J-1,I,NP,LP,LK,NY,LY);CALL RANGE(IA-J-1,NC,LC,LK,NG,
238. 1,IG);CALL MULFRA(NZ,LZ,NY,LY,NG,LG,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
239. IF(IT.EQ.1)STOP
240. CALL RANGE(I,NR,LR,LK,NG,LG);CALL SOUFRA(NY,LY,NG,LG,NZ,LZ,N1,N2,N
241. 1,N4,N5,N6,N7,LK,IT);IF(IT.EQ.1)STOP
242. CALL REPAR(IA-J-1,I-1,NH,LH,LK,NG,LG);CALL ADDFRA(NZ,LZ,NY,LY,NG,L
243. 1,IG,N1,N2,N3,N4,N5,N6,N7,LK,IT);IF(IT.EQ.1)STOP
244. 25 CALL MISE(I,NZ,LZ,LK,NR,LR)
245. 220 WRITE(IN,300)JA,J
246. DO 28 I=0,J
247. CALL EGAL(1,IA-J,I,NH,LH,LK,IA-J,I,NHR,LHR)
248. CALL EGAL(1,IA-J,I,NP,LP,LK,IA-J,I,NPR,LPR)
249. CALL RANGE(I,NR,LR,LK,NX,LX);CALL MACHIN(IA-J,I,NX,LA,LK,NH,LN
250. CALL RANGE(I,NT,LT,LK,NY,LY);CALL MACHIN(IA-J,I,NY,LY,LK,NP,LP)
251. 28 CONTINUE
252. DO 88 I=0,J
253. CALL RANGE(I,NT,LT,LK,NX,LX);CALL QUOTIE(NX,LX,LK,Q1)
254. CALL ECR(NX,LK,RZ,LR1);WRITE(IN,320)Q1,I;CALL ECR(LX,LK,RZ,LR1)
255. 88 WRITE(IN,1000);WRITE(IN,625)JA,J
256. DO 89 I=0,J-1
257. CALL RANGE(I,NR,LR,LK,NX,LX);CALL QUOTIE(NX,LX,LK,Q1)
258. CALL ECR(NX,LK,RZ,LR1);WRITE(IN,320)Q1,I;CALL ECR(LX,LK,RZ,LR1)
259. 89 WRITE(IN,1000)
260. C APPEL A ECRAT
261. CALL ECRAT(NT,LT,NR,LR,IA,J,N1,N2,N3,N4,N5,N6,N7,N8,NX,LX,NY,LY,N
262. 1,LZ,LK,LR1,RZ);GO TO 501
263. 201 CALL STU(0,LK,NX);CALL STO(1,LK,LX)
264. CALL MACHIN(IA-J,J,NX,LX,LK,NP,LP);CALL MACHIN(IA-J,J,NX,LX,LK,NH,

```



```

265. 1(LH);CALL MACHIN(IA-J,J,NY,LX,LV,NPR,LPR)
266. CALL MACHIN(IA-J,J,NX,LX,LK,NHR,LHR)
267. 501 IF(J.LT.NL)J=J+1;GO TO 370
268. NL=NL+1;GO TO 360
269. END

```

```

1. SUBROUTINE EGRAT(NS,LS,NR,LR,IA,J,N1,N2,N3,N4,N5,N6,N7,N8,NX,LX,NY
2. 1,LY,NZ,LZ,LK,LR1,RX)
3. C*****
4. C SOUS PROGRAMME CALCULANT L'APPROXIMANT DE PADE
5. C*****
6. C*****CARTES SUSCEPTIBLES D'ETRE MODIFIEES PAR L'UTILISATEUR*****
7. COMMON NM,NI,P/T/NCR(-10:10,0:10,11),LCR(-10:10,0:10,11),LBR(-10:1
8. 10,0:10,11),NR(-10:10,0:10,11),LQ(-10:10,0:10,11),LE(-10:10,0:10,1
9. 1),NE(-10:10,0:10,11),NBR(-10:10,0:10,11),LC(-10:10,11),NC(-10:10,1
10. 11),NDF(-10:10,0:10)/S/NA(10),KA(10),NRA(10)/BLOK1/TBASE,LM,IN,KN
11. DIMENSIONNT(0:20,11),LT(0:20,11),NW(0:20,11),LW(0:20,11),RX(LR1)
12. DIMENSIONNS(-10:10,LK),LS(-10:10,LK),NR(-10:10,LK),LR(-10:10,LK)
13. C*****FIN*****
14. DIMENSIONNX(LK),LX(LK),NY(LK),LY(LK),NZ(LK),LZ(LK)
15. DIMENSIONN1(LK),N2(LK),N3(LK),N4(LK),N5(LK),N6(LK),N7(LK),N8(LK)
16. DO 4 I=0,J-1
17. CALL RANGE(J-1-I,NR,LR,LK,NX,LX);CALL MISE1(I,NX,LX,LK,NW,LW)
18. 4 CONTINUE;IB=IA-1;WRITE(IN,300)IB,J
19. DO 5 I=1,J
20. CALL RANGE(J-I,NS,LS,LK,NX,LX);CALL MISE1(I,NX,LX,LK,NR,LR)
21. CALL RANGE1(J-I,NW,LW,LK,NZ,LZ);CALL MISE1(J-I,NZ,LZ,LK,NT,LT)
22. 5 CONTINUE;CALL STO(1,LK,LX);CALL MISE(0,LX,LX,LK,NR,LR)
23. IF(IA-J.EQ.0)GO TO 8
24. DO 6 I=0,IA-1
25. CALL STO(0,LK,NX);CALL STO(1,LK,LX);M1=MAX(0,J+1-IA+I);M2=MIN(I,J)
26. IF(I.GE.IA-J)CALL RANGE1(I-IA+J,NW,LW,LK,NX,LX)
27. CALL MISE1(I,NX,LX,LK,NT,LT)
28. DO 6 K=M1,M2
29. CALL RANGE(I-K,NC,LC,LK,NX,LX);CALL RANGE(K,NR,LR,LK,NY,LY)
30. CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
31. IF(IT.EQ.1)STOP
32. CALL RANGE1(I,NT,LT,LK,NX,LX);CALL ADDFRA(NY,LY,NX,LX,NZ,LZ,N1,N2,
33. 1N3,N4,N5,N6,N7,LK,IT);IF(IT.EQ.1)STOP
34. 6 CALL MISE1(I,NY,LY,LK,NT,LT)
35. 8 WRITE(IN,205)
36. 205 FORMAT(/,2X,'NUMERATEUR',/)
37. DO 1 I=0,IA-1
38. CALL RANGE1(I,NT,LT,LK,NX,LX);CALL QUOTIE(NX,LX,LK,Q1)
39. CALL ECR(NX,LK,RX,LR1);WRITE(IN,2003)Q1,I;CALL ECR(LX,LK,RX,LR1)
40. 1 WRITE(IN,1000);WRITE(IN,206)
41. 206 FORMAT(/,2X,'DENOMINATEUR',/)
42. DO 2 I=0,J
43. CALL RANGE(I,NR,LR,LK,NX,LX);CALL QUOTIE(NX,LX,LK,Q1)
44. CALL ECR(NX,LK,RX,LR1);WRITE(IN,2003)Q1,I;CALL ECR(LX,LK,RX,LR1)
45. 2 WRITE(IN,1000);RETURN
46. 1000 FORMAT(/)
47. 2003 FORMAT('-----=',D22,15,'x+',I3)
48. 300 FORMAT(/,3X,'APPROXIMANT DE PADE('',I3,'',',',I3,'')')
49. END

```

VIII.4. SOUS-PROGRAMMES DE CALCUL DES APPROXIMANTS DE PADE EN DEUX POINTS

On rappelle que $P_{k,m}(x) = P_m^{(k-2m)}(x)$. Nous écrirons $Q_{k,m}(x) = Q_m^{(k-2m)}(x)$ pour utiliser les notations de PADERA.

* * * PA2POI * * * *

Appel : CALL PA2POI(C1,C2, N, N1, N2,..., N7,N8, NX, LX, NY, LY, NZ, LZ, NG, LG, LK, LR)

Paramètres : C1, C2 et N sont définis comme dans PADPOI.

LK et LR sont des entiers calculés par INI et les autres paramètres sont des T.E.P. de longueur LK.

Tous les paramètres sont des paramètres d'entrée.

Sous-programmes utilisés : QUDERA, ESTRA, DETECT et IMPARA.

Description : Même description que PADPOI.

* * * IMPARA * * * *

Appel : CALL IMPARA(NS, LS, NT, LT, L, LR, K2, J, NX, LX, NY, LY)

Paramètres : Les NS(I, •) et LS(I, •) (resp. NT(I, •) et LT(I, •)) pour $0 \leq I \leq J$, sont des T.E.P. de longueur L, ils représentent les numérateurs et dénominateurs des coefficients de x^I dans le polynôme $P_{K2,J}(x)$ (resp. $Q_{K2,J}(x)$).

L et LR sont les entiers déterminés par INI.

Les autres paramètres sont des T.E.P. de longueur L.

Tous les paramètres sont des paramètres d'entrée.

Description : Même description que IMPADE.

APPROXIMANT DE PADE EN DEUX POINTS(7, 7)

NUMERATEUR

$$\begin{array}{r} + 0 \\ \hline + 1 \end{array} = .0000000000000000+00 \quad x^{**} \quad 0$$

$$\begin{array}{r} - 10 \\ \hline + 1 \end{array} = -.1000000000000000+02 \quad x^{**} \quad 1$$

$$\begin{array}{r} - 515970084653 \\ \hline + 40943545841 \end{array} = -.1250198827567880+02 \quad x^{**} \quad 2$$

$$\begin{array}{r} + 331268342927 \\ \hline + 163774183364 \end{array} = .2022714057384320+01 \quad x^{**} \quad 3$$

$$\begin{array}{r} - 478738110 \\ \hline + 40943545841 \end{array} = -.1109263922228740-01 \quad x^{**} \quad 4$$

$$\begin{array}{r} + 0 \\ \hline + 1 \end{array} = .0000000000000000+00 \quad x^{**} \quad 5$$

$$\begin{array}{r} + 0 \\ \hline + 1 \end{array} = .0000000000000000+00 \quad x^{**} \quad 6$$

DENOMINATEUR

$$\begin{array}{r} + 1 \\ \hline + 1 \end{array} = .1000000000000000+01 \quad x^{**} \quad 0$$

$$\begin{array}{r} - 2553434982282573918107 \\ \hline + 33164272131210 \end{array} = -.7699354812251710+08 \quad x^{**} \quad 1$$

$$\begin{array}{r} - 12932692311490392537087 \\ \hline + 132657088524840 \end{array} = -.9748964382762300+08 \quad x^{**} \quad 2$$

$$\begin{array}{r} + 15204197425 \\ \hline + 81887091682 \end{array} = .1856727002107230+00 \quad x^{**} \quad 3$$

$$\begin{array}{r} + 2875667125 \\ \hline + 81887091682 \end{array} = .3511746559723180-01 \quad x^{**} \quad 4$$

$$\begin{array}{r} + 4931412120 \\ \hline + 40943545841 \end{array} = .1204441876907930+00 \quad x^{**} \quad 5$$

$$\begin{array}{r} + 20182671545 \\ \hline + 163774183364 \end{array} = .1252347561162470+00 \quad x^{**} \quad 6$$

$$\begin{array}{r} - 797896850 \\ \hline + 40943545841 \end{array} = -.1948773203714570-01 \quad x^{**} \quad 7$$



EXEMPLE D'APPEL DES SOUS-PROGRAMMES

Le programme suivant :

```

1. C*****
2. C   PROGRAMME PRINCIPAL POUR LE CALCUL DES APPROXIMANTS DE PADE EN
3. C   DEUX POINTS
4. C*****
5. C*****CARTES POUVANT ETRE CHANGEES PAR L'UTILISATEUR*****
6. COMMON NM,NI,P/T/MCR(-10:10,0:10,11),LCR(-10:10,0:10,11),LBR(-10:1
7. 10,0:10,11),NCR(-10:10,0:10,11),LQ(-10:10,0:10,11),LE(-10:10,0:10,1
8. 1),NE(-10:10,0:10,11),NRR(-10:10,0:10,11),LC(-10:10,11),NC(-10:10,1
9. 11),NDF(-10:10,0:10)/S/NA(10),KA(10),NPA(10)/BLOK1/TRASE,LM,IN,KN
10. NM=2147483647;NI=108;P=1.D-14;IBASE=100000000;IN=108;LM=20;KN=105
11. DIMENSIONN1(11),N2(11),N3(11),N4(11),N5(11),N6(11),N7(11),N8(11)
12. DIMENSIONNX(11),LY(11),LZ(11),NZ(11),LX(11),NY(11),NG(11),LG(11)
13. DIMENSIONC1(-10:10),C2(-10:10)
14. C*****FIN*****
15. READ(KN,10)N
16. READ(KN,20)(C1(I),I=-N,N)
17. READ(KN,20)(C2(I),I=-N,N)
18. 10 FORMAT(15)
19. 20 FORMAT(5F15,8)
20. DO 1 J=-N,N
21. 1 WRITE(IN,70)J,C1(J),J,C2(J)
22. 70 FORMAT(5X,'C1(','I3,')=' ,D22,15,25X,'C2(','I3,')=' ,D22,15)
23. CALL INI(LK,LR);CALL PA2POI(C1,C2,N,N1,N2,N3,N4,N5,N6,N7,N8,NX,LX,
24. 1NY,LY,NZ,LZ,NG,LG,LK,LR);STOP
25. END

```

lit N et les coefficients $\overline{c_i}$ et $\overline{c_i^*}$ des séries formelles, avant de faire appel aux sous-programmes qui calculent les approximants de Padé en deux points.

Pour N = 7 et les coefficients $\overline{c_i}$ et $\overline{c_i^*}$ suivants :

C1(-7)= .0000000000000000+00	C2(-7)= .0000000000000000+00
C1(-6)= .5000000000000000+01	C2(-6)= .1500000000000000+02
C1(-5)= .2000000000000000+01	C2(-5)= .7000000000000000+02
C1(-4)= .1000000000000000+03	C2(-4)= .0000000000000000+00
C1(-3)= .2000000000000000+00	C2(-3)= .8000000000000000+00
C1(-2)= .0000000000000000+00	C2(-2)= .0000000000000000+00
C1(-1)= .0000000000000000+00	C2(-1)= .0000000000000000+00
C1(0)= .0000000000000000+00	C2(0)= .0000000000000000+00
C1(1)= .1000000000000000+01	C2(1)= .1100000000000000+02
C1(2)= .2000000000000000+01	C2(2)= .1000000000000000+02
C1(3)= .1000000000000000+02	C2(3)= .7000000000000000+01
C1(4)= .0000000000000000+00	C2(4)= .0000000000000000+00
C1(5)= .0000000000000000+00	C2(5)= .0000000000000000+00
C1(6)= .0000000000000000+00	C2(6)= .0000000000000000+00
C1(7)= .0000000000000000+00	C2(7)= .0000000000000000+00

où C1 (resp. C2) est le tableau des coefficients $\overline{c_i}$ (resp. $\overline{c_i^*}$).


```

1.      SUBROUTINE PA2POI(C1,C2,NL,N1,N2,N3,N4,N5,N6,N7,N8,NX,LX,NY,LY,NZ,
2.      1LZ,NG,LG,LK,LR1)
3.      C*****
4.      C      CALCUL DES POLYNOMES ORTHOGONAUX REGULIERS ET LEURS ASSOCIES, EN VUE DU
5.      C      CALCUL DES APPROXIMANTS DE PADE EN DEUX POINTS
6.      C*****
7.      C***** CARTES SUSCEPTIBLES D'ETRE MODIFIEES PAR L'UTILISATEUR*****
8.      COMMON NH,NI,P/T/NCR(-10:10,0:10,11),LCR(-10:10,0:10,11),LRR(-10:1
9.      10,0:10,11),NRQ(-10:10,0:10,11),LC(-10:10,0:10,11),LE(-10:10,0:10,11
10.     1),NE(-10:10,0:10,11),NRR(-10:10,0:10,11),LC(-10:10,1:11),NR(-10:10,1
11.     11),NDF(-10:10,0:10)/S/NA(20),KA(20),NRA(20)/KLOK1/4ASF,LM,IN,KM
12.     DIMENSIONNR(-10:10,0:10,11),NT(-10:10,0:10,11),LR(-10:10,0:10,11)
13.     DIMENSIONNR(-10:10,0:10,11),LT(-10:10,0:10,11),LW(-10:10,0:10,11)
14.     DIMENSIONNH(-10:10,0:10,11),LH(-10:10,0:10,11),NP(-10:10,0:10,11),
15.     1LP(-10:10,0:10,11),NPR(-10:10,0:10,11),LPR(-10:10,0:10,11)
16.     DIMENSIONC1(-10:NL),C2(-10:NL)
17.      C*****FIN*****
18.     DIMENSIONNX(LK),LX(LK),NY(LK),LY(LK),NZ(LK),LZ(LK),NG(LK),LG(LK)
19.     DIMENSIONN1(LK),N2(LK),N3(LK),N4(LK),N5(LK),N6(LK),N7(LK),N8(LK)
20.     DEPS=1./NH;IF(NH.LE.0)WRITE(IN,500)STOP
21.     500 FORMAT(3X,'ERREUR LA VALEUR DE NH DOIT ETRE STRICT SUP A ZERO')
22.     DO 36 I=-NL,NL
23.     INA=0;IDA=1;IT=0;IF(ABS(C1(I)).GE.DEPS)CALL RAT(C1(I),INA,IDA,R,IT
24.     1)
25.     IF(IT.EQ.1)STOP
26.     CALL STO(INA,LK,NX);CALL STO(IDA,LK,LX)
27.     INA=0;IDA=1;IT=0;IF(ABS(C2(I)).GE.DEPS)CALL RAT(C2(I),INA,IDA,R,IT
28.     1)
29.     IF(IT.EQ.1)STOP
30.     CALL STO(INA,LK,NY);CALL STO(IDA,LK,LY)
31.     CALL SOUFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,LK,IT)
32.     IF(IT.EQ.1)STOP
33.     CALL QUOTIE(NZ,LZ,LK,Q);WRITE(IN,800)I,Q
34.     36 CALL MISE(I,NZ,LZ,LK,NC,LC)
35.     800 FORMAT(5X,'D(',I3,')=' ,D22,15)
36.     CALL QUADER(NL,-NL,0,LA,N1,N2,N3,N4,N5,N6,N7,N8,NX,LX,NY,LY,NZ,LZ,
37.     1LK,LR1)
38.      C*****
39.      C
40.      C      CALCUL DES POLYNOMES ORTHOGONAUX REGULIERS DE LA DEMI-TABLE SUPERIEURE
41.      C      ET LEURS ASSOCIES
42.      C
43.      C*****
44.     IA=-NL;ML=NL;CALL STO(0,LK,NX);CALL STO(1,LK,LX)
45.     CALL MACHIN(IA,0,NX,LX,LK,NPR,LPR);CALL MACHIN(IA,0,NX,LX,LK,NH,LH
46.     1);CALL MACHIN(IA,0,LX,LX,LK,NP,LP);IA=IA+1
47.     100 CALL STO(0,LK,NX);CALL STO(1,LK,LX)
48.     CALL MACHIN(IA,0,NX,LX,LK,NPR,LPR);CALL MACHIN(IA,0,NX,LX,LK,NH,LH
49.     1);CALL MACHIN(IA,0,LX,LX,LK,NP,LP);J=1
50.     340 IF(IA.GT.-1)GO TO 360
51.     K2=IA+J;JA=IA-J;IF(NDF(IA-J,J).EQ.0)WRITE(IN,350)JA,J;GO TO 101
52.     CALL STO(0,LK,NX);CALL STO(1,LK,LX)
53.     CALL MISE(J,LX,LX,LK,NT,LT);CALL MISE(J,NX,LX,LK,NR,LR)
54.     IF(NDF(IA-J+1,J-1).EQ.0)GO TO 102
55.     CALL REPAR(IA-J,J,NQ,LQ,LK,NX,LX);CALL REPAR(IA-J,0,NP,LP,LK,NY,LY
56.     1);CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,NR,LK,IT)
57.     IF(IT.EQ.1)STOP
58.     NZ(1)=-NZ(1);CALL MISE(0,NZ,LZ,LK,NT,LT)
59.     CALL STO(0,LK,NX);CALL STO(1,LK,LX)
60.     CALL MISE(0,NX,LX,LK,NR,LR);IF(J.EQ.1)GO TO 110
61.     DO 2 I=1,J-1
62.     CALL STO(0,LK,NX);CALL STO(1,LK,LX);CALL MISE(I,NX,LX,LK,NR,LR)
63.     CALL REPAR(IA-J,J,NQ,LQ,LK,NG,LG);CALL REPAR(IA-J,I,NP,LP,LK,NY,LY
64.     1);CALL MULFRA(NZ,LZ,NG,LG,NY,LY,N1,N2,N3,N4,N5,N6,N7,NR,LK,IT)
65.     IF(IT.EQ.1)STOP
66.     CALL REPAR(IA-J+1,I-1,NP,LP,LK,NY,LY)
67.     CALL SOUFRA(NX,LX,NY,LY,NZ,LZ,N1,N2,N3,N4,N5,N6,N7,LK,IT)
68.     IF(IT.EQ.1)STOP
69.     2 CALL MISE(I,NX,LX,LK,NT,LT)
70.     110 IF(K2.LT.0)GO TO 120
71.     IF((K2.EQ.0).OR.(K2.EQ.1))GO TO 135
72.     K5=K2/2
73.     DO 3 I=1,K5
74.     IF(NDF(IA-J,I-1).NE.0)GO TO 130
75.     3 CONTINUE;GO TO 135
76.     350 FORMAT(3X,'LE POLYNOME ',2X,I3,2X,I3,3X,'NON O.R.')
77.     130 CALL REPAR(IA-J,J,NQ,LQ,LX,NX,LX);NY(1)=-NY(1)
78.     CALL REPAR(IA-J,0,NH,LH,LK,NY,LY);CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1
79.     1,N2,N3,N4,N5,N6,N7,NR,LK,IT);IF(IT.EQ.1)STOP
80.     CALL MISE(0,NZ,LZ,LK,NR,LR)
81.     DO 9 I=1,J-1
82.     CALL REPAR(IA-J,J,NQ,LQ,LX,NG,LG);CALL REPAR(IA-J,I,NH,LH,LK,NY,LY
83.     1);CALL MULFRA(NZ,LZ,NG,LG,NY,LY,N1,N2,N3,N4,N5,N6,N7,NR,LK,IT)
84.     IF(IT.EQ.1)STOP
85.     CALL REPAR(IA-J+1,I-1,NH,LH,LK,NY,LY)
86.     CALL SOUFRA(NX,LX,NY,LY,NZ,LZ,N1,N2,N3,N4,N5,N6,N7,LK,IT)
87.     IF(IT.EQ.1)STOP
88.     9 CALL MISE(I,NX,LX,LK,NR,LR);GO TO 120

```

```

69. C*****
90. C   CAS OU LE POLYNOME EST SUR L'EST D'UN BLOC
91. C*****
92. 102 CALL DETECT(IA-J,1,J,LA,L);NR1=NRA(L);L3=NA(L)
93.   IS=IA-J-L3+NR1+1;IF(IA-J.GT.-NL)GO TO 103
94.   DO 6 I=0,IS
95.     CALL REPAR(L3-NR1-1+I,J,NBR,LBR,LK,NX,LX)
96.   6 CALL MISE(IS-I,NX,LX,LK,NW,LW);CALL STO(1,LK,LX)
97.     CALL MISE(IS+1,LX,LX,LK,NW,LW)
98.     DO 30 I=0,J-1
99.       CALL STO(I,LK,NX);CALL STO(1,LK,LX);CALL MISE(I,NX,LX,LK,NR,LR)
100.      M4=MAX(0,I+IS+1-J);M5=MIN(IS+1,I);CALL MISE(I,NX,LX,LK,NT,LT)
101.      DO 30 K=M4,M5
102.        CALL REPAR(IA-J,I-K,NP,LP,LK,NX,LX);CALL RANGE(K,NW,LW,LK,NY,LY)
103.        CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
104.        IF(IT.EQ.1)STOP
105.        CALL RANGE(I,NT,LT,LK,NX,LX)
106.        CALL ADDFRA(NY,LY,NZ,LZ,NX,LX,N1,N2,N3,N4,N5,N6,N7,LK,IT)
107.        IF(IT.EQ.1)STOP
108.      30 CALL MISE(I,NY,LY,LK,NT,LT)
109.      DO 16 I=0,J-IS-2
110.        CALL REPAR(IA-J,I,NCR,LCR,LK,NX,LX);CALL REPAR(IA-J,I,NPR,LPR,LK,N
111.        1Y,LY);CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
112.        IF(IT.EQ.1)STOP
113.        CALL RANGE(I,NT,LT,LK,NX,LX)
114.        CALL ADDFRA(NY,LY,NZ,LZ,NX,LX,N1,N2,N3,N4,N5,N6,N7,LK,IT)
115.        IF(IT.EQ.1)STOP
116.      16 CALL MISE(I,NY,LY,LK,NT,LT);IF(K2.LT.0)GO TO 120
117. C*****
118. C   CALCUL DE L'ASSOCIE EN UTILISANT LA DEFINITION
119. C*****
120. 135 DO 40 IS=0,J-1
121.     CALL STO(0,LK,NX);CALL STO(1,LK,LX);CALL MISE(IS,NX,LX,LK,NR,LR)
122.     DO 40 I6=0,I5
123.       CALL RANGE(IS-I6,NT,LT,LK,NX,LX);CALL RANGE(-1-I6,NC,LC,LK,NY,LY)
124.       CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
125.       IF(IT.EQ.1)STOP
126.       CALL RANGE(IS,NR,LR,LK,NX,LX);NZ(1)=-NZ(1)
127.       CALL ADDFRA(NY,LY,NZ,LZ,NX,LX,N1,N2,N3,N4,N5,N6,N7,LK,IT)
128.       IF(IT.EQ.1)STOP
129.     40 CALL MISE(IS,NY,LY,LK,NR,LR);GO TO 120
130. 103 CALL REPAR(IA-J-1,J,NE,LE,LK,NX,LX);IF(IA-J.LE.L3-NR1-1)NX(1)=-NX(
131. 11)
132.     DO 4 I=0,J-1
133.       CALL STO(0,LK,NX);CALL STO(1,LK,LX);CALL MISE(I,NX,LX,LK,NR,LR)
134.       CALL REPAR(IA-J,I,NP,LP,LK,NY,LY)
135.       CALL MULFRA(NZ,LZ,NY,LY,NX,LX,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
136.       IF(IT.EQ.1)STOP
137.       CALL REPAR(IA-J-1,I,NP,LP,LK,NY,LY);CALL ADDFRA(NG,LG,NZ,LZ,NY,LY,
138.       1N1,N2,N3,N4,N5,N6,N7,LK,IT);IF(IT.EQ.1)STOP
139.     4 CALL MISE(I,NG,LG,LK,NT,LT);IF(K2.LT.0)GO TO 120
140.     K5=K2/2;IF(K5.LE.0)GO TO 135
141.     DO 5 I=1,K5
142.       IF(NDF(IA-J,J-I),NE.0)GO TO 47
143.     5 CONTINUE;GO TO 135
144. 47 DO 42 I=0,J-1
145.     CALL REPAR(IA-J,I,NH,LH,LK,NY,LY)
146.     CALL MULFRA(NZ,LZ,NY,LY,NX,LX,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
147.     IF(IT.EQ.1)STOP
148.     CALL REPAR(IA-J-1,I,NH,LH,LK,NY,LY);CALL ADDFRA(NG,LG,NZ,LZ,NY,LY,
149.     1N1,N2,N3,N4,N5,N6,N7,LK,IT);IF(IT.EQ.1)STOP
150. 42 CALL MISE(I,NG,LG,LK,NR,LR)
151. 120 WRITE(IN,300)JA,J
152.     DO 8 I=0,J
153.       CALL EGAL(1,IA-J,I,NP,LP,LK,IA-J,I,NPR,LPR)
154.       CALL RANGE(I,NR,LR,LK,NX,LX);CALL MACHIN(IA-J,I,NX,LX,LK,NH,LH)
155.       CALL RANGE(I,NT,LT,LK,NY,LY);CALL MACHIN(IA-J,I,NY,LY,LK,NP,LP)
156.     8 CONTINUE;IF(K2.LT.0)GO TO 301
157. C   APPEL DE IMPARA
158.     CALL IMPARA(NR,LR,NT,LT,LK,LR1,K2,J,NX,LX,NY,LY);GO TO 301
159. 101 CALL STO(0,LK,NX);CALL STO(1,LK,LX)
160.     CALL MACHIN(IA-J,J,NX,LX,LK,NP,LP);CALL MACHIN(IA-J,J,NX,LX,LK,NH,
161.     1LH);CALL MACHIN(IA-J,J,NX,LX,LK,NPR,LPR)
162. 301 IF(IA-J.GT.-NL)J=J+1;GO TO 340
163.     IA=IA+1;GO TO 100
164. C*****
165. C
166. C   CALCUL DES POLYNOMES ORTHOGONAUX REGULIERS DE LA DEMI-TABLE INFERIEURE
167. C   ET LEURS ASSOCIES
168. C
169. C*****
170. 360 IF(IA.GT.NL)RETURN
171.     IF(IA.EQ.1)NL=NL
172.     CALL STO(0,LK,NX);CALL STO(1,LK,LX)
173.     CALL MACHIN(IA,J,NX,LX,LK,NPR,LPR);CALL MACHIN(IA,J,NX,LX,LK,NH,LH
174.     1);CALL MACHIN(IA,J,LX,LX,LK,NP,LP);J=1
175. 370 JA=IA-J;IF(NDF(JA,J),EQ.0)WRITE(IN,350)JA,J;GO TO 201
176.     CALL STO(0,LK,NX);CALL STO(1,LK,LX)

```

```

177. CALL MISE(J,LX,LV,LK,NT,LT); CALL MISE(J,NX,LA,LY,NR,LR); *2=IA+J
178. IF(NDF(IA-J+1,J-1,ED,0)GO TO 202
179. CALL REPAR(IA-J,J,NQ,LQ,LK,NX,LX); CALL REPAR(IA-J,0,NP,LP,LK,NY,LY
180. 1); CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,NR,LK,IT)
181. IF(IT,ED,1)STOP
182. NZ(1)=-NZ(1); CALL MISE(0,NZ,LZ,LK,NT,LT); CALL STO(0,LK,NX)
183. CALL STO(1,LK,LX); CALL MISE(0,NX,LY,LK,NR,LR); IF(J,EG,1)GO TO 240
184. DO 22 I=1,J-1
185. CALL STO(0,LK,NX); CALL STO(1,LK,LX); CALL MISE(1,NX,LX,LK,NR,LR)
186. CALL REPAR(IA-J,J,NQ,LQ,LK,NX,LG); CALL REPAR(IA-J,I,IP,LP,LK,NY,LY
187. 1); CALL MULFRA(NZ,LZ,NQ,LG,NY,LY,N1,N2,N3,N4,N5,N6,N7,NR,LK,IT)
188. IF(IT,ED,1)STOP
189. CALL REPAR(IA-J+1,I-1,NP,LP,LK,NY,LY)
190. CALL SOUFRA(NX,LX,NY,LY,NZ,LZ,N1,N2,N3,N4,N5,N6,N7,LK,IT)
191. IF(IT,ED,1)STOP
192. 22 CALL MISE(I,NX,LX,LK,NT,LT)
193. 240 IF(IA-J,GT,0)GO TO 220
194. IF(IA-J,ED,0)GO TO 260
195. IF((IA-J,ED,-1),AND,(J,ED,1))GO TO 260
196. IF(IA-J,LT,-2)K5=K2/2; GO TO 68
197. DO 55 I=1,J-1
198. IF(NDF(IA-J,J-1),NE,0)GO TO 54
199. 55 CONTINUE; GO TO 260
200. 68 DO 59 I=1,K5
201. IF(NDF(IA-J,J-1),NE,0)GO TO 54
202. 59 CONTINUE; GO TO 260
203. 54 CALL REPAR(IA-J,J,NQ,LQ,LK,NX,LX); NX(1)=-NX(1)
204. CALL REPAR(IA-J,0,NH,LH,LK,NY,LY); CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1
205. 1,N2,N3,N4,N5,N6,N7,NR,LK,IT); IF(IT,ED,1)STOP
206. CALL MISE(0,NZ,LZ,LK,NR,LR)
207. DO 23 I=1,J-1
208. CALL REPAR(IA-J,J,NQ,LQ,LK,NX,LG); CALL REPAR(IA-J,I,NH,LH,LK,NY,LY
209. 1); CALL MULFRA(NZ,LZ,NQ,LG,NY,LY,N1,N2,N3,N4,N5,N6,N7,NR,LK,IT)
210. IF(IT,ED,1)STOP
211. CALL REPAR(IA-J+1,I-1,NH,LH,LK,NY,LY)
212. CALL SOUFRA(NX,LX,NY,LY,NZ,LZ,N1,N2,N3,N4,N5,N6,N7,LK,IT)
213. IF(IT,ED,1)STOP
214. 23 CALL MISE(I,NX,LX,LK,NR,LR); GO TO 220
215. C*****
216. C CALCUL DE L'ASSOCIE EN UTILISANT LA DEFINITION
217. C*****
218. 260 DO 43 I=0,J-1
219. CALL STO(0,LK,NX); CALL STO(1,LK,LX); CALL MISE(I,NX,LX,LK,NR,LR)
220. DO 43 I=0,J-1-I
221. CALL RANGE(L+1,I,NT,LT,LK,NX,LX); CALL RANGE(L,NC,LC,LK,NY,LY)
222. CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
223. IF(IT,ED,1)STOP
224. CALL RANGE(I,NR,LR,LK,NX,LX)
225. CALL ADDFRA(NY,LY,NX,LX,NZ,LZ,N1,N2,N3,N4,N5,N6,N7,LK,IT)
226. IF(IT,ED,1)STOP
227. 43 CALL MISE(I,NY,LY,LK,NR,LR); GO TO 220
228. C*****
229. C CAS OU LE POLYNOME EST SUR L'EST D'UN BLOC
230. C*****
231. 202 CALL DETECT(IA-J,I,J,LA,L); NR1=NRA(L); L3=NA(L)
232. IS=IA-J-L3+NR1+1; IF(IA-J,GT,-NL)GO TO 203
233. DO 666 I=0,IS
234. CALL REPAR(L3-NR1+1+I,J,NBR,LBR,LK,NX,LX)
235. 666 CALL MISE(IS-I,NX,LX,LK,NW,LW); CALL STO(1,LK,LX)
236. CALL MISE(IS+1,LX,LX,LK,NW,LW)
237. DO 630 I=0,J-1
238. CALL STO(0,LK,NX); CALL STO(1,LK,LX); CALL MISE(I,NX,LX,LK,NR,LR)
239. M4=MAX(0,I+IS+1-J); M5=MIN(IS+1,I); CALL MISE(I,NX,LX,LK,NT,LT)
240. DO 630 K=M4,M5
241. CALL REPAR(IA-J,I-K,NP,LP,LK,NX,LX); CALL RANGE(K,NW,LW,LK,NY,LY)
242. CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
243. IF(IT,ED,1)STOP
244. CALL RANGE(I,NT,LT,LK,NX,LX)
245. CALL ADDFRA(NY,LY,NZ,LZ,NX,LX,N1,N2,N3,N4,N5,N6,N7,LK,IT)
246. IF(IT,ED,1)STOP
247. 630 CALL MISE(I,NY,LY,LK,NT,LT)
248. DO 616 I=0,J-IS-2
249. CALL REPAR(IA-J,J,NCR,LCR,LK,NX,LX); CALL REPAR(IA-J,I,NPR,LPR,LK,N
250. 1Y,LY); CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
251. IF(IT,ED,1)STOP
252. CALL RANGE(I,NT,LT,LK,NX,LX)
253. CALL ADDFRA(NY,LY,NZ,LZ,NX,LX,N1,N2,N3,N4,N5,N6,N7,LK,IT)
254. IF(IT,ED,1)STOP
255. 616 CALL MISE(I,NY,LY,LK,NT,LT); GO TO 260
256. 203 CALL REPAR(IA-J-1,J,NE,LE,LK,NX,LX); IF(IA-J,LE,L3-NR1-1)NX(1)=-NX(
257. 1)
258. DO 24 I=0,J-1
259. CALL STO(0,LK,NX); CALL STO(1,LK,LX); CALL MISE(I,NX,LX,LK,NR,LR)
260. CALL REPAR(IA-J,I,NP,LP,LK,NY,LY)
261. CALL MULFRA(NZ,LZ,NY,LY,NX,LX,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
262. IF(IT,ED,1)STOP
263. CALL REPAR(IA-J-1,I,NP,LP,LK,NY,LY); CALL ADDFRA(NQ,LG,NZ,LZ,NY,LY,
264. 1,N1,N2,N3,N4,N5,N6,N7,LK,IT); IF(IT,ED,1)STOP

```

BUS
FILE

```

265.      24 CALL MISE(I,NG,LG,LK,NT,LT);IF(IA-J.GT.0)GO TO 220
266.      IF(IA-J.LT.-2)K5=K2/2;GO TO 80
267.      DO 82 I=1,J-1
268.      IF(NDF(IA-J,J-1),NE.0)GO TO 38
269.      82 CONTINUE;GO TO 260
270.      80 DO 32 I=1,K5
271.      IF(NDF(IA-J,J-1),NE.0)GO TO 38
272.      32 CONTINUE;GO TO 260
273.      38 DO 35 I=0,J-1
274.      CALL REPAR(IA-J,I,NH,LH,LK,NY,LY)
275.      CALL MULFRA(NZ,LZ,NY,LY,NX,LX,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
276.      IF(IT.EQ.1)STOP
277.      CALL REPAR(IA-J-1,I,NH,LH,LK,NY,LY);CALL ADDFRA(NG,LG,NZ,LZ,NY,LY,
278.      1N1,N2,N3,N4,N5,N6,N7,LK,IT);IF(IT.EQ.1)STOP
279.      35 CALL MISE(I,NG,LG,LK,NR,LR)
280.      220 WRITE(IN,300)JA,J
281.      DO 28 I=0,J
282.      CALL EGAL(1,IA-J,I,NP,LP,LK,IA-J,I,NPR,LPR)
283.      CALL RANGE(I,NR,LR,LK,NX,LX);CALL MACHIN(IA-J,I,NX,LX,LK,NH,LK)
284.      CALL RANGE(I,NT,LT,LK,NY,LY);CALL MACHIN(IA-J,I,NY,LY,LK,NP,LP)
285.      28 CONTINUE;IF(IA-J.GT.0)GO TO 401
286.      C APPEL DE IMPARA
287.      CALL IMPARA(NR,LR,NT,LT,LK,LR1,K2,J,NX,LX,NY,LY);GO TO 401
288.      201 CALL STO(0,LK,NX);CALL STO(1,LK,LX)
289.      CALL MACHIN(IA-J,J,NX,LX,LK,NP,LP);CALL MACHIN(IA-J,J,NX,LX,LK,NH,
290.      1LH);CALL MACHIN(IA-J,J,NX,LX,LK,NPR,LPR)
291.      401 IF(J.LT.ML)J=J+1;GO TO 370
292.      IA=IA+1;ML=ML-1;GO TO 360
293.      300 FORMAT(1X,'(',I3,',',I3,')=' ,2X,'ASSOCIE',8X,'POLYNOME O.R. ')
294.      END

```

```

1.      SUBROUTINE IMPARA(NS,LS,NT,LT,L,LR,K2,J,NX,LX,NY,LY)
2.      C*****
3.      C SOUS PROGRAMME D'IMPRESSION DE L'APPROXIMANT DE PADE EN DEUX POINTS
4.      C*****
5.      C*****CARTES SUSCEPTIBLES D'ETRE MODIFIEES PAR L'UTILISATEUR*****
6.      DIMENSIONNS(-10:10,L),LS(-10:10,L),NT(-10:10,L),LT(-10:10,L)
7.      DIMENSIONNX(L),LX(L),NY(L),LY(L),RX(160)
8.      MAT=108
9.      C*****FIN*****
10.     WRITE(MAT,20)K2,J;WRITE(MAT,500);I1=J-1
11.     500 FORMAT(2X,/,,'NUMERATEUR',/)
12.     DO 1 I=0,J-1
13.     CALL RANGE(I1,NS,LS,L,NX,LX);CALL QUOTIE(NX,LX,L,Q)
14.     CALL ECR(NX,L,RX,LR);WRITE(MAT,2003)Q,I;CALL ECR(LY,L,RX,LR)
15.     WRITE(MAT,1000)
16.     1 I1=I1-1;WRITE(MAT,700);I2=J
17.     DO 2 I=0,J
18.     CALL RANGE(I2,NT,LT,L,NX,LX);CALL QUOTIE(NX,LX,L,Q)
19.     CALL ECR(NX,L,RX,LR);WRITE(MAT,2003)Q,I;CALL ECR(LY,L,RX,LR)
20.     WRITE(MAT,1000)
21.     2 I2=I2-1;RETURN
22.     1000 FORMAT(/)
23.     700 FORMAT(2X,/,,'DENOMINATEUR',/)
24.     2003 FORMAT('-----=' ,D22,15,1X,'X**',I3)
25.     20 FORMAT(//,3X,'APPROXIMANT DE PADE EN DEUX POINTS(' ,I3,',',I3,')')
26.     END

```

VIII.5. SOUS-PROGRAMME DE CALCUL DES APPROXIMANTS DE SERIES DE FONCTIONS

Dans le sous-programme SRIFON que l'on propose les polynômes sont notés comme dans PADERA.

* * * SRIFON * * * *

Appel : CALL SRIFON(NL, N1, N2, N3, N4, N5, N6, N7, N8, NX, NY, LY, LX, NZ, LZ, NG, LG, LK, LR)

Paramètres : NL est défini comme dans SERFON. LK et LR sont les entiers calculés par INI. Les autres paramètres sont des T.E.P. de longueur LK.

Tous les paramètres sont des paramètres d'entrée.

Sous-programmes utilisés : QUDERA, DETECT, ESTRA, GAUSS, BAIRNE, RACINE et MATRI.

Description : Ce sous-programme calcule les approximants $[p-1/p]$ des séries de fonction f_n .

EXEMPLE D'APPEL DU SOUS-PROGRAMME SRIFON

On propose un programme qui lit les valeurs de N et des c_i , et fait appel au sous-programme SRIFON.

```

1.  C*****
2.  C   PROGRAMME PRINCIPAL-CALCUL DES APPROXIMANTS D'UNE SERIE DE FONCTIONS
3.  C*****
4.  C*****CARTES POUVANT ETRE CHANGEES PAR L'UTILISATEUR*****
5.  COMMON NM,PI,P/T/NCR(-10:10,0:10,11),LCR(-10:10,0:10,11),LBR(-10:1
6.  10,0:10,11),NR(-10:10,0:10,11),Lo(-10:10,0:10,11),LE(-10:10,0:10,11
7.  1),NE(-10:10,0:10,11),NBR(-10:10,0:10,11),LC(-10:10,11),NC(-10:10,1
8.  11),NDF(-10:10,0:10)/S/NA(10),KA(10),NRA(10)/BLOK1/IBASE,LM,IN,KN
9.  DIMENSIONN1(11),N2(11),N3(11),N4(11),N5(11),N6(11),N7(11),N8(11)
10. DIMENSIONNX(11),NY(11),NZ(11),NG(11),LX(11),LY(11),LZ(11),LG(11)
11. DIMENSIONC(0:10)
12. N1=2147483647;N2=108;P=1,D=14;IBASE=100000000;IN=108;LM=80;KN=105
13. C*****FIN*****
14. READ(KN,20)N
15. READ(KN,10)(C(I),I=0,N)
16. 20 FORMAT(I3)
17. 10 FORMAT(10F15.8)
18. DEPS=1./NM;CALL INI(L,LR)
19. DO 1 I=0,N
20.   INA=0;IDA=1;IT=0;IF(ABS(C(I)).GE,DEPS)CALL RAT(C(I),INA,IDA,R,IT)
21.   IF(IT.EQ,1)STOP
22.   CALL STO(INA,L,NX);CALL STO(IDA,L,LX)
23. 1 CALL MISE(I,NX,LX,L,NC,LC)
24. CALL SRIFON(N,N1,N2,N3,N4,N5,N6,N7,N8,NX,NY,LY,LX,NZ,LZ,NG,LG,L,LR
25. 1);STOP
26. END

```


où les numéros de gauche sont les indices des diagonales et les + (resp. 0) donnent les positions des polynômes orthogonaux réguliers (resp. non orthogonaux réguliers),

donne l'allure de la table P des polynômes orthogonaux.

Donnons quelques approximants de la forme $[p-1/p]_{f_n}$.

Ces approximants sont donnés sous la même forme que dans l'essai numérique de la partie VI.

$$[1/2]_{f_1} =$$

.5000000000000000-04	.1443375672974020+10	*G(0, .1000000000000040-04	-.1732050807568900-04)
.5000000000000000-04	-.1443375672974020+10	*G(0, .1000000000000040-04	.1732050807568900-04)

$$[2/3]_{f_0} =$$

-.6249999999999250+14	.3608439182435020+14	*G(0, .1000000000000040-04	-.1732050807568900-04)
-.6249999999999250+14	-.3608439182435020+14	*G(0, .1000000000000040-04	.1732050807568900-04)
.1249999999999850+15	.1171875000000000-01	*G(0, .0000000000000000+00	.0000000000000000+00)

$$[3/4]_{f_0} =$$

.6000000000000000-04	.0000000000000000+00	*G(0, .0000000000000000+00	.0000000000000000+00)
.1000000000000000-03	.0000000000000000+00	*G(1, .0000000000000000-00	.0000000000000000+00)
.2500000000000000+00	.0000000000000000+00	*G(2, .0000000000000000+00	.0000000000000000+00)
.1666666666666667+00	.0000000000000000+00	*G(3, .0000000000000000+00	.0000000000000000+00)

• Cette racine se justifie par le fait que $P_3^{(0)}(x) = x P_2^{(1)}(x) - q_3^{(0)} P_2^{(0)}(x)$

avec

$$\begin{aligned}
 &= 1499999995 \\
 &----- \\
 &+ 15624999999981250075000000499997
 \end{aligned}
 = G(0, 3) = -.9599999968001170-23$$

LE SOUS-PROGRAMME SRIFON

```

1.      SUBROUTINE SRIFON(NL,N1,N2,N3,N4,N5,N6,N7,N8,NX,NY,LY,LX,NZ,LZ,NG,
2.      1LG,LK,LP)
3.      C*****
4.      C      SOUS PROGRAMME-CALCUL DES APPROXIMANTS D'UNE SERIE DE FONCTIONS
5.      C*****
6.      C*****CARTES SUSCEPTIBLES D'ETRE MODIFIEES PAR L'UTILISATEUR*****
7.      COMMON /M,NH,I,P/T/NCR(-10:10,0:10,11),LCR(-10:10,0:10,11),LBR(-10:1
8.      10,0:10,11),NCR(-10:10,0:10,11),LQ(-10:10,0:10,11),LE(-10:10,0:10,11
9.      1),NE(-10:10,0:10,11),NBR(-10:10,0:10,11),LC(-10:10,11),NC(-10:10,1
10.     11),NDF(-10:10,0:10)/S/NA(20),KA(20),NRA(20)/BLOK1/IRASF,LM,IX,KN
11.     COMPLEXX(0:10),AM1(0:10,0:10),LAND(0:10)
12.     DIMENSIONT(0:10),Y(0:10),TQ(0:10),NT(-10:10,11),LT(-10:10,11)
13.     DIMENSIONNH(LK),LX(LK),LY(LK),LZ(LK),NG(LK),LG(LK)
14.     DIMENSIONN1(LK),N2(LK),N3(LK),N4(LK),N5(LK),N6(LK),N7(LK),N8(LK)
15.     1,LPR(-10:10,0:10,11)
16.     C*****FIN*****
17.     DIMENSIONNH(LK),LX(LK),LY(LK),LZ(LK),NG(LK),LG(LK)
18.     DIMENSIONN1(LK),N2(LK),N3(LK),N4(LK),N5(LK),N6(LK),N7(LK),N8(LK)
19.     C      APPEL A QUADERA
20.     CALL QUADERA(NL,0,0,LA,N1,N2,N3,N4,N5,N6,N7,N8,NX,LX,NY,LY,NZ,LZ,LK
21.     1,LRI)
22.     C*****
23.     C
24.     C      CALCUL DES POLYNOMES ORTHOGONAUX REGULIERS DE LA DEMI-TABLE SUPERIEURE
25.     C      ET DES APPROXIMANTS ASSOCIES A CES POLYNOMES
26.     C*****
27.     IA=0;ML=(NL+1)/2;CALL STO(0,LK,NX);CALL STO(1,LK,LY)
28.     CALL MACHIN(IA,0,NX,LX,LK,NPR,LPR);CALL MACHIN(IA,0,LX,LY,LK,NP,LP
29.     1);IA=1
30.     100 IF(IA.GT.ML)IA=IA-1;GO TO 390
31.     CALL MACHIN(IA,0,NX,LX,LK,NPR,LPR);CALL MACHIN(IA,0,LX,LY,LK,NP,LP
32.     1);J=1
33.     340 JH=IA-J;IF(NDF(IA-J,J).EQ.0)J1=J-1;WRITE(IN,350)J1,J,JA;GO TO 101
34.     CALL STO(1,LK,LX);CALL MISE(J,LX,LK,NT,LT)
35.     IF(NDF(IA-J+1,J-1).EQ.0)GO TO 102
36.     CALL REPAR(IA-J,J,NQ,LQ,LK,NX,LX);CALL REPAR(IA-J,0,NP,LP,LK,NY,LY
37.     1);CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
38.     IF(IT.EQ.1)STOP
39.     NZ(1)=-NZ(1);CALL MISE(0,NZ,LZ,LK,NT,LT);IF(J.EQ.1)GO TO 120
40.     DO 2 I=1,J-1
41.     CALL REPAR(IA-J,I,NP,LP,LK,NX,LX);CALL REPAR(IA-J,J,NQ,LQ,LK,NY,LY
42.     1);CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
43.     IF(IT.EQ.1)STOP
44.     CALL REPAR(IA-J+1,I-1,NP,LP,LK,NX,LX)
45.     CALL SOUFRA(NY,LY,NX,LX,N7,LZ,N1,N2,N3,N4,N5,N6,N7,LK,IT)
46.     IF(IT.EQ.1)STOP
47.     2 CALL MISE(I,NY,LY,LK,NT,LT);GO TO 120
48.     102 CALL DETECT(IA-J,1,J,LA,L);IA1=NRA(L);IA2=NAL(L)
49.     IS=IA-J-IA2+IA1+1;IF(IA-J.GT.0)GO TO 103
50.     DO 6 I=0,IS
51.     CALL REPAR(IA2-IA1-1+I,J,NBR,LBR,LK,NX,LX)
52.     6 CALL MISE(IS-I,NX,LX,LK,NW,LW);CALL STO(1,LK,LX)
53.     CALL MISE(IS+1,LX,LX,LK,NW,LW)
54.     DO 30 I=0,J-1
55.     M5=MIN(IS+1,I)
56.     CALL STO(0,LK,NX);CALL STO(1,LK,LX);M4=MAX(0,I+1+IS-J)
57.     CALL MISE(I,NX,LX,LK,N7,LT)
58.     DO 30 K=M4,M5
59.     CALL REPAR(IA-J,I-K,NP,LP,LK,NX,LX);CALL RANGE(X,NW,LW,LK,NY,LY)
60.     CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
61.     IF(IT.EQ.1)STOP
62.     CALL RANGE(I,NT,LT,LK,NX,LX)
63.     CALL ADDFRA(NY,LY,NZ,LZ,NX,LX,N1,N2,N3,N4,N5,N6,N7,LK,IT)
64.     IF(IT.EQ.1)STOP
65.     30 CALL MISE(I,NY,LY,LK,NT,LT)
66.     DO 16 I=0,J-IS-2
67.     CALL REPAR(IA-J,J,NCR,LCR,LK,NX,LX);CALL REPAR(IA-J,I,NPR,LPR,LK,
68.     1Y,LY);CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
69.     IF(IT.EQ.1)STOP
70.     CALL RANGE(I,NT,LT,LK,NX,LX)
71.     CALL ADDFRA(NY,LY,NZ,LZ,NX,LX,N1,N2,N3,N4,N5,N6,N7,LK,IT)
72.     IF(IT.EQ.1)STOP
73.     16 CALL MISE(I,NY,LY,LK,NT,LT);GO TO 120
74.     103 CALL REPAR(IA-J-1,J,NE,LE,LK,NX,LX);IF(IA-J.LE.IA2-IA1-1)NZ(1)=-NZ
75.     1(1)
76.     DO 4 I=0,J-1
77.     CALL REPAR(IA-J,I,NP,LP,LK,NY,LY)
78.     CALL MULFRA(NZ,LZ,NY,LY,NX,LX,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
79.     IF(IT.EQ.1)STOP
80.     CALL REPAR(IA-J-1,I,NP,LP,LK,NY,LY);CALL ADDFRA(NQ,LQ,NZ,LZ,NY,LY,
81.     1N1,N2,N3,N4,N5,N6,N7,LK,IT);IF(IT.EQ.1)STOP
82.     4 CALL MISE(I,NQ,LQ,LK,NT,LT)
83.     120 J1=J-1;WRITE(IN,350)J1,J,JA
84.     DO 3 I=1,J
85.     CALL EGAL(1,IA-J,I,NP,LP,LK,IA-J,I,NPR,LPR)
86.     CALL RANGE(I,NT,LT,LK,NY,LY);CALL MACHIN(IA-J,I,NY,LY,LK,NP,LP)
87.     3 CALL QUOTIE(NY,LY,LK,T(I))
88.     C      CALCUL DES RACINES DES POLYNOMES

```

BUS
FILE

```

89. CALL BAIRNE(T,J,X,IQ,I2)
90. 1002 FORMAT(10X,2(D22,15),5X,'*G(',13,'.',2(D22,15),')')
91. C DETERMINATION DES COEFFICIENTS DE LA MATRICE
92. CALL MATRI(X,IQ,J,I2,AM1)
93. DO 46 I=0,J-1
94. CALL RANGE(IA-J+I,NC,LC,LK,NX,LX);CALL QUOTIE(NX,LX,LK,Y(I))
95. 46 CONTINUE
96. C RESOLUTION DU SYSTEME D'EQUATIONS
97. CALL GAUSS(AM1,Y,J,LANO);IK=0
98. DO 50 I=0,I2
99. DO 52 L=0,IQ(I)-1
100. 52 WRITE(IN,1002)LANO(IK+L),L,X(I)
101. 50 IK=IK+IQ(I);GO TO 301
102. 101 CALL STO(0,LK,NX);CALL STO(1,LK,LX)
103. CALL MACHIN(IA-J,J,NX,LX,LK,NP,LP);CALL MACHIN(IA-J,J,NX,LX,LK,NPR
104. 1,LP)
105. 301 IF(J,LT,IA)J=J+1;GO TO 340
106. IA=IA+1;GO TO 100
107. 350 FORMAT(//,3X,'APPROXIMANT(',I2,'/',I2,')F',I2,'EST DANS UN BLOC')
108. 300 FORMAT(//,3X,'APPROXIMANT(',I2,'/',I2,')F',I2,/)
109. C*****
110. C
111. C CALCUL DES POLYNOMES ORTHOGONAUX REGULIERS DE LA DEMI-TABLE INFERIEURE
112. C ET DES APPROXIMANTS ASSOCIES A CES POLYNOMES
113. C*****
114. 390 IF(MOD(NL,2),NE,0)ML=ML-1
115. 360 IA=IA+1;IF(IA,GT,NL)RETURN
116. CALL MACHIN(IA,0,NX,LX,LK,NPR,LP);CALL MACHIN(IA,0,LX,LA,LK,NP,LP
117. 1)
118. 370 JA=IA-J;IF(NDF(IA-J,J),EQ,0)J1=J-1;WRITE(IN,300)J1,J,JA;GO TO 201
119. CALL STO(1,LK,LX);CALL MISE(J,LX,LX,LK,NT,IT)
120. IF(NDF(IA-J+1,J-1),EQ,0)GO TO 202
121. CALL REPAR(IA-J,J,NQ,LO,LK,NX,LX);CALL REPAR(IA-J,0,NP,LP,LK,NY,LY
122. 1);CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
123. IF(IT,EQ,1)STOP
124. NZ(1)=-NZ(1);CALL MISE(0,NZ,LZ,LK,NT,LT);IF(J,EQ,1)GO TO 220
125. DO 22 I=0,J-1
126. CALL REPAR(IA-J,I,NP,LP,LK,NX,LX);CALL REPAR(IA-J,J,NQ,LO,LK,NY,LY
127. 1);CALL MULFRA(NZ,LZ,NX,LX,NY,LY,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
128. IF(IT,EQ,1)STOP
129. CALL REPAR(IA-J+1,I-1,NP,LP,LK,NX,LX)
130. CALL SOUPRA(NY,LY,NX,LX,NZ,LZ,N1,N2,N3,N4,N5,N6,N7,LK,IT)
131. IF(IT,EQ,1)STOP
132. 22 CALL MISE(1,NY,LY,LK,NT,LT);GO TO 220
133. 202 CALL DETECT(IA-J,1,J,LA,L);IA1=NRA(L);IA2=NA(L)
134. CALL REPAR(IA-J-1,J,NE,LE,LK,NY,LX);IF(IA-J,LE,IA2-IA1-1)NX(1)=-NX
135. 1(1)
136. DO 24 I=0,J-1
137. CALL REPAR(IA-J,I,NP,LP,LK,NY,LY)
138. CALL MULFRA(NZ,LZ,NY,LY,NX,LX,N1,N2,N3,N4,N5,N6,N7,N8,LK,IT)
139. IF(IT,EQ,1)STOP
140. CALL REPAR(IA-J-1,I,NP,LP,LK,NY,LY);CALL ADDFRA(NQ,LO,NZ,LZ,NY,LY,
141. 1,N1,N2,N3,N4,N5,N6,N7,LK,IT);IF(IT,EQ,1)STOP
142. 24 CALL MISE(1,NQ,LO,LK,NT,LT)
143. 220 J1=J-1;WRITE(IN,300)J1,J,JA
144. DO 28 I=0,J
145. CALL EGAL(1,IA-J,I,NP,LP,LK,IA-J,I,NPR,LP)
146. CALL RANGE(I,NT,LT,LK,NY,LY);CALL MACHIN(IA-J,I,NY,LY,LK,NP,LP)
147. 28 CALL QUOTIE(NY,LY,LK,T(I))
148. C CALCUL DES RACINES DES POLYNOMES
149. CALL BAIRNE(T,J,X,IQ,I2)
150. C DETERMINATION DES COEFFICIENTS DE LA MATRICE
151. CALL MATRI(X,IQ,J,I2,AM1)
152. DO 56 I=0,J-1
153. CALL RANGE(IA-J+I,NC,LC,LK,NX,LX);CALL QUOTIE(NX,LX,LK,Y(I))
154. 56 CONTINUE
155. C RESOLUTION DU SYSTEME D'EQUATIONS
156. CALL GAUSS(AM1,Y,J,LANO);IK=0
157. DO 60 I=0,I2
158. DO 62 L=0,IQ(I)-1
159. 62 WRITE(IN,1002)LANO(IK+L),L,X(I)
160. 60 IK=IK+IQ(I);GO TO 901
161. 201 CALL STO(0,LK,NX);CALL STO(1,LK,LX)
162. CALL MACHIN(IA-J,J,NX,LX,LK,NP,LP);CALL MACHIN(IA-J,J,NX,LX,LK,NPR
163. 1,LP)
164. 901 IF(J,LT,ML)J=J+1;GO TO 370
165. ML=ML+1;GO TO 360
166. END

```

NEUVIEME PARTIE

COMPARAISONS ET CONCLUSION

Dans cette partie on compare les résultats obtenus avec les arithmétiques réelle double précision et rationnelle étendue sur un exemple pour lequel on calcule la table q_d , les polynômes orthogonaux réguliers et leurs associés, et les approximants de Padé.

L'arithmétique rationnelle étendue est une arithmétique exacte. Une erreur peut être commise chaque fois que l'on calculera le quotient d'un rationnel, cette erreur est toujours bien contrôlée puisqu'elle ne dépend que des erreurs faites en prenant les arrondis des numérateur et dénominateur et de l'erreur sur l'opération (voir description du sous-programme QUOTIE).

Dans l'arithmétique réelle double précision il faut tenir compte de la propagation des erreurs de calcul puisque tout calcul est entâché d'une erreur due aux erreurs antérieures et aux erreurs sur les opérations.

Pour $NL = 10$ et les coefficients c_i suivants :

```

C( 0)= .4000000000000000+01
C( 1)= .5000000000000000+01
C( 2)= .1500000000000000+02
C( 3)= .3500000000000000+02
C( 4)= .9900000000000000+02
C( 5)= .2750000000000000+03
C( 6)= .7950000000000000+03
C( 7)= .2315000000000000+04
C( 8)= .6819500000000000+04
C( 9)= .2019500000000000+05
C( 10)= .6007500000000000+05
  
```

la table P des polynômes orthogonaux a l'allure suivante :

N= 0							
N= 1	+						
N= 2		+	+				
N= 3			+	+	+		
N= 4				+	+	+	+
N= 5					+	+	+
N= 6						+	0
N= 7							0
N= 8							
N= 9							
N= 10							
N= 11							

Dans le tableau ci-dessus les numéros de gauche sont les indices des diagonales et les + (resp. 0) donnent les positions des polynômes orthogonaux réguliers (resp. non orthogonaux réguliers)

A) Comparons les résultats de la colonne d'indice 4 de la table qd

1) Avec l'arithmétique rationnelle étendue on obtient :

INDICE DE COLONNE= 4

```

+ 11
-----=Q( 0, 4)= .1571428571428570+01
+ 7

+ 0
-----=E( 0, 4)= .0000000000000000+00
+ 189

+ 35
-----=Q( 1, 4)= .1296296296296300+01
+ 27

+ 0
-----=E( 1, 4)= .0000000000000000+00
+ 1755

+ 81
-----=Q( 2, 4)= .1246153846153850+01
+ 65

+ 0
-----=E( 2, 4)= .000000 000000000+00
+ 2405

+ 39
-----=Q( 3, 4)= .1054054054054050+01
+ 37

```

L'erreur sur les quotients est majorée par la précision machine,
 10^{-15} pour l'IRIS 80.

2) Avec l'arithmétique réelle double précision on a :

INDICE DE COLONNE= 4

	VALEURS DE Q(N,J)	VALEURS DE E(N,J)
N= 0	.1571428571428530+01	.1065814103640150-12
N= 1	.1296296296296370+01	.0000000000000000+00
N= 2	.1246153846153780+01	.0000000000000000+00
N= 3	.1054054054053950+01	



L'erreur commise sur les valeurs des q_4 et des e_4 est majorée par $e_4^{(0)}$.

On fait un test sur la nullité des valeurs des colonnes e et chaque fois que l'on a une valeur nulle on dit que l'on est à l'ouest d'un bloc q_d , ce test est donc fondamental pour la détection des blocs.

Si le test sur la nullité des valeurs des colonnes e ne pose aucun problème avec l'arithmétique rationnelle étendue, il n'en est pas de même avec l'arithmétique réelle double précision puisqu'il faut tenir compte de la propagation des erreurs de calcul. On considère comme nulles les valeurs des colonnes e inférieures à un seuil ϵ que l'on se donne, il n'est pas toujours facile de se donner ϵ puisque la propagation des erreurs dépend non seulement de l'algorithme mais aussi des valeurs des c_i . Un mauvais choix de la valeur de ϵ fait que l'on a des blocs en trop (resp. en moins) ou trop grand (resp. trop petit) si ϵ est trop grand (resp. trop petit). On prendra ϵ égal à la précision machine ($10^{-\alpha}$ si l'ordinateur travaille avec α chiffres) pour tout test de nullité des coefficients de la série formelle.

Dans l'exemple que nous considérons si on prend $\epsilon < e_4^{(0)}$ on aura $P_5^{(0)}$ orthogonal régulier, ce qui est faux.

B) Comparons les polynômes à l'Ouest du bloc et leurs associés

1) Avec l'arithmétique rationnelle étendue on a :

Les polynômes sont donnés sous la même forme que dans l'essai numérique de VIII.3.

$$P_4^{(0)}(x)$$

$$\begin{array}{r} = 6 \\ \hline \text{-----} - ,6000000000000000+01 \text{ x** } 0 \\ + 1 \end{array}$$

$$\begin{array}{r} + 5 \\ \hline \text{-----} ,5000000000000000+01 \text{ x** } 1 \\ + 1 \end{array}$$

$$\begin{array}{r} + 5 \\ \hline \text{-----} ,5000000000000000+01 \text{ x** } 2 \\ + 1 \end{array}$$

$$\begin{array}{r} = 5 \\ \hline \text{-----} -,5000000000000000+01 \text{ x** } 3 \\ + 1 \end{array}$$

$$\begin{array}{r} + 1 \\ \hline \text{-----} ,1000000000000000+01 \text{ x** } 4 \\ + 1 \end{array}$$

Les $P_4^{(i)}(x)$ pour $i=1, 2$ et 3 sont identiques à $P_4^{(0)}(x)$ et les associés $Q_4^{(0)}(x)$ pour $i=0, 1, 2$ et 3 sont :

$$Q_4^{(0)}(x) =$$

$$\frac{+ 5}{+ 1} \text{-----} = .5000000000000000+01 \text{ x}^{++} 0$$

$$\frac{+ 10}{+ 1} \text{-----} = .1000000000000000+02 \text{ x}^{++} 1$$

$$\frac{- 15}{+ 1} \text{-----} = -.1500000000000000+02 \text{ x}^{++} 2$$

$$\frac{+ 4}{+ 1} \text{-----} = .4000000000000000+01 \text{ x}^{++} 3$$

$$Q_4^{(1)}(x) =$$

$$\frac{+ 24}{+ 1} \text{-----} = .2400000000000000+02 \text{ x}^{++} 0$$

$$\frac{- 15}{+ 1} \text{-----} = -.1500000000000000+02 \text{ x}^{++} 1$$

$$\frac{- 10}{+ 1} \text{-----} = -.1000000000000000+02 \text{ x}^{++} 2$$

$$\frac{+ 5}{+ 1} \text{-----} = .5000000000000000+01 \text{ x}^{++} 3$$

$$\frac{+ 35}{+ 1} \text{-----} = .3500000000000000+02 \text{ x}^{++} 0$$

$$Q_4^{(2)}(x) =$$

$$\frac{+ 30}{+ 1} \text{-----} = .3000000000000000+02 \text{ x}^{++} 0$$

$$\frac{- 1}{+ 1} \text{-----} = -.1000000000000000+01 \text{ x}^{++} 1$$

$$\frac{- 40}{+ 1} \text{-----} = -.4000000000000000+02 \text{ x}^{++} 2$$

$$\frac{+ 15}{+ 1} \text{-----} = .1500000000000000+02 \text{ x}^{++} 3$$



$Q_4^{(3)}(x)$

```

+ 90
-----
+ 1      ,9000000000000000+02 x** 0

+ 45
-----
+ 1      -,4500000000000000+02 x** 1

+ 76
-----
+ 1      -,7600000000000000+02 x** 2

+ 35
-----
+ 1      ,3500000000000000+02 x** 3

```

Les quotients des coefficients des polynômes $P_4^{(i)}(x)$ pour $i = 0, 1, 2$ et 3 sont exacts, il en est de même pour les quotients des coefficients des $Q_4^{(i)}(x)$ pour $i = 0, 1, 2$ et 3.

2) Avec l'arithmétique réelle double précision les résultats sont :

Les polynômes sont donnés sous la même forme que dans l'essai numérique de III.

$$Q_4^{(0)}(x)$$

$$P_4^{(0)}(t)$$

```
.49999999999999990+01X** 0      -.59999999999999850+01T** 0
-.99999999999999990+01X** 1      .49999999999999990+01T** 1
-.14999999999999990+02X** 2      .499999999999999970+01T** 2
.40000000000000000+01X** 3      -.49999999999999960+01T** 3
.00000000000000000+00X** 4      .10000000000000000+01T** 4
```

$$Q_4^{(1)}(x)$$

$$P_4^{(1)}(t)$$

```

.240000000000000120+02x** 0      -.600000000000000340+01T** 0
-.14999999999999980+02x** 1      .50000000000000010+01T** 1
-.10000000000000030+02x** 2      .500000000000000240+01T** 2
.50000000000000000+01x** 3      -.50000000000000070+01T** 3
.00000000000000000+00x** 4      .10000000000000000+01T** 4

```

$$Q_4^{(2)}(x)$$

$$P_4^{(2)}(t)$$

```

      .299999999999986D+02X** 0      -.599999999999967D+01T** 0
-   .1000000000000165D+01X** 1      .500000000000008D+01T** 1
-   .399999999999937D+02X** 2      .499999999999972D+01T** 2
    .150000000000000D+02X** 3      -.49999999999993D+01T** 3
    .000000000000000D+00X** 4      .100000000000000D+01T** 4

```

$$Q_4^{(3)}(x)$$

$$P_4^{(3)}(t)$$

[illegible]

L'erreur maximale sur les coefficients des polynômes $P_4^{(i)}(t)$ (resp.

$Q_4^{(i)}(x)$ pour $i = 0, 1, 2$ et 3 , est $5,6 \cdot 10^{-13}$ (resp. $3,9 \cdot 10^{-12}$). Les coefficients soulignés sont ceux qui sont entâchés de la plus grande erreur.



C) Comparons les approximants de Padé $[i/4]_f$ pour $i = 3, 4, 5$ et 6

1) Avec l'arithmétique rationnelle étendue les résultats sont :

Les approximants $[i/4]_f$ pour $i = 3, 4$ et 5 sont identiques à l'approximant $[6/4]_f$.

APPROXIMANT DE PADE(6, 4)

NUMERATEUR

$$\begin{array}{r} + 4 \\ \hline + 1 \end{array} = .4000000000000000+01 \times 10^0$$

$$\begin{array}{r} - 15 \\ \hline + 1 \end{array} = -.1500000000000000+02 \times 10^1$$

$$\begin{array}{r} + 10 \\ \hline + 1 \end{array} = .1000000000000000+02 \times 10^2$$

$$\begin{array}{r} + 5 \\ \hline + 1 \end{array} = .5000000000000000+01 \times 10^3$$

$$\begin{array}{r} + 0 \\ \hline + 1 \end{array} = .0000000000000000+00 \times 10^4$$

$$\begin{array}{r} + 0 \\ \hline + 1 \end{array} = .0000000000000000+00 \times 10^5$$

$$\begin{array}{r} + 0 \\ \hline + 1 \end{array} = .0000000000000000+00 \times 10^6$$

DENOMINATEUR

$$\begin{array}{r} + 1 \\ \hline + 1 \end{array} = .1000000000000000+01 \times 10^0$$

$$\begin{array}{r} - 5 \\ \hline + 1 \end{array} = -.5000000000000000+01 \times 10^1$$

$$\begin{array}{r} + 5 \\ \hline + 1 \end{array} = .5000000000000000+01 \times 10^2$$

$$\begin{array}{r} + 5 \\ \hline + 1 \end{array} = .5000000000000000+01 \times 10^3$$

$$\begin{array}{r} - 6 \\ \hline + 1 \end{array} = -.6000000000000000+01 \times 10^4$$

Les quotients des coefficients des approximants $[i/4]_f$ pour $i = 3, 4, 5$ et 6 sont exacts.

2) Avec l'arithmétique réelle double précision on a les résultats suivants :

APPROXIMANT DE PADE(3, 4)

NUMERATEUR	DENOMINATEUR
.4000000000000000+01T** 0	.1000000000000000+01T** 0
-.1499999999999999+02T** 1	-.4999999999999999+01T** 1
.9999999999999999+01T** 2	.4999999999999999+01T** 2
.4999999999999999+01T** 3	.4999999999999999+01T** 3
	-.5999999999999999+01T** 4

APPROXIMANT DE PADE(4, 4)

NUMERATEUR	DENOMINATEUR
.4000000000000000+01T** 0	.1000000000000000+01T** 0
-.1500000000000000+02T** 1	-.5000000000000000+01T** 1
.1000000000000000+02T** 2	.5000000000000000+01T** 2
.5000000000000000+01T** 3	.5000000000000000+01T** 3
-.1385558334732200-12T** 4	-.6000000000000000+01T** 4

APPROXIMANT DE PADE(5, 4)

NUMERATEUR	DENOMINATEUR
.4000000000000000+01T** 0	.1000000000000000+01T** 0
-.1499999999999999+02T** 1	-.4999999999999999+01T** 1
.9999999999999999+01T** 2	.4999999999999999+01T** 2
.5000000000000000+01T** 3	.5000000000000000+01T** 3
-.6750155989720950-13T** 4	-.5999999999999999+01T** 4
.2735589532676390-12T** 5	

APPROXIMANT DE PADE(6, 4)

NUMERATEUR	DENOMINATEUR
.4000000000000000+01T** 0	.1000000000000000+01T** 0
-.1499999999999999+02T** 1	-.4999999999999999+01T** 1
<u>.9999999999999999+01T** 2</u>	.4999999999999999+01T** 2
.4999999999999999+01T** 3	.5000000000000000+01T** 3
.2060573933704290-12T** 4	-.5999999999999999+01T** 4
.2913225216616410-12T** 5	
.1918465386552270-12T** 6	



L'erreur maximale sur les coefficients des approximants de Padé $[i/4]_f$ pour $i = 3, 4, 5$ et 6 est $1,15 \cdot 10^{-12}$, elle est commise sur le coefficient souligné.

Sur les résultats ci-dessus on remarque l'élévation parasite des degrés des numérateurs de $[4/4]_f$, $[5/4]_f$ et $[6/4]_f$, ces approximants devraient être identiques à $[3/4]_f$.

Cet essai numérique, comme tous les autres que nous avons faits, montre combien il est plus prudent d'utiliser une arithmétique exacte pour le calcul de la table q_d , le calcul des polynômes orthogonaux réguliers et de leurs associés, et pour le calcul des approximants.

Dans le cas d'un ordinateur dont la capacité limite pour les entiers est assez petite on est obligé d'utiliser une arithmétique étendue, évidemment elle est plus coûteuse en mémoires et en temps par rapport à une arithmétique réelle. Elle demande autant de fois plus de mémoires que la longueur des tableaux mémorisant les entiers. Cette technique de faire les opérations sur les tableaux nécessite plus de temps :

1,610 contre 0,189 minute pour la compilation,
0,771 contre 0,163 minute pour l'édition de lien,
et 0,343 contre 0,031 minute pour l'exécution

pour l'exemple que nous venons de considérer.

REFERENCES

- [1] DRAUX A.
"Polynômes orthogonaux formels. Applications".
Lecture Note in Mathematics. 947, Springer-Verlag, Heidelberg 1983.
- [2] AMAMOU H.
"Conception et réalisation de logiciels d'arithmétique".
Thèse de 3ème Cycle. Lille I, 1982.
- [3] BREZINSKI C.
"Outlines of PADE Approximation".
Publication A.N.O. n° 71, Mai 1982, Laboratoire d'Informatique
U.E.R. d'I.E.E.A. Lille I.
- [4] DRAUX A.
"Approximants de Padé en deux points dans une Algèbre non commutative".
Publication A.N.O. Lille I, (à paraître).
- [5] HENRICI P.
"The quotient-difference algorithm".
NBS Appl. Math. Series, 49(1958), pp. 23-46.
- [6] RUTISHAUSER H.
"Der Quotienten-Differenzen Algorithmus".
Birkhäuser Verlag, Basel, 1957.

